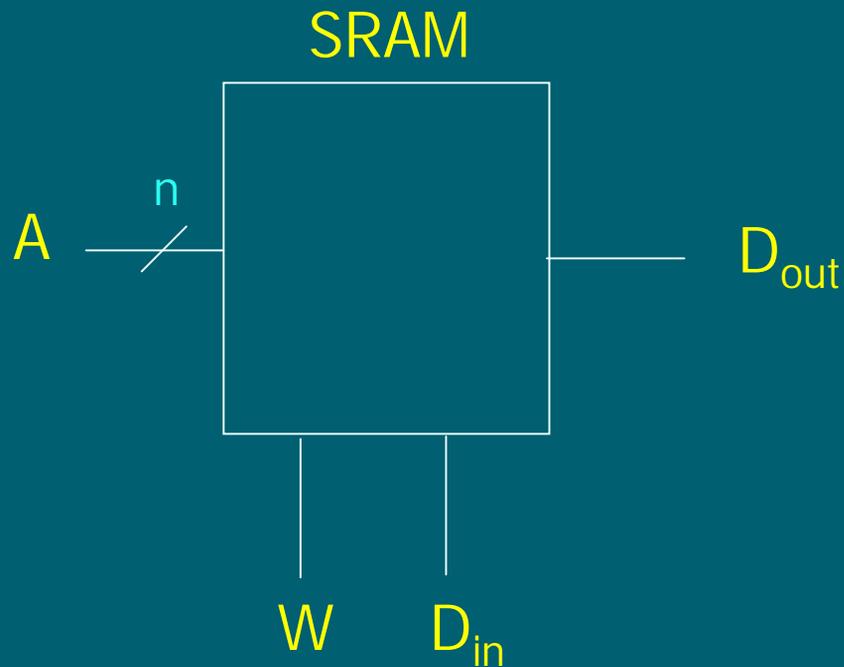


12.3 SRAM



Statischer Speicher - Schaltsymbol



Ein N-Bit SRAM

Sei $n \in \mathbf{N}$, $N = 2^n$



Ein N-Bit SRAM

Sei $n \in \mathbf{N}$, $N = 2^n$

Ein N-Bit statischer Speicher oder **SRAM**
(static random access memory) hat:



Ein N-Bit SRAM

Sei $n \in \mathbf{N}$, $N = 2^n$

Ein N-Bit statischer Speicher oder **SRAM**

(static random access memory) hat:

- n Eingänge $A = (A_{n-1}, \dots, A_0)$ „Adresse“



Ein N-Bit SRAM

Sei $n \in \mathbf{N}$, $N = 2^n$

Ein N-Bit statischer Speicher oder **SRAM**

(static random access memory) hat:

- n Eingänge $A = (A_{n-1}, \dots, A_0)$ „Adresse“
- Dateneingang D_{in} , Datenausgang D_{out}



Ein N-Bit SRAM

Sei $n \in \mathbf{N}$, $N = 2^n$

Ein N-Bit statischer Speicher oder **SRAM**

(static random access memory) hat:

- n Eingänge $A = (A_{n-1}, \dots, A_0)$ „Adresse“
- Dateneingang D_{in} , Datenausgang D_{out}
- Kontrollsignal W „write“



N-Bit SRAM (ff)



N-Bit SRAM (ff)

- Der Speicher enthält N Speicherzellen L_0, \dots, L_{N-1} , die je ein Bit speichern können.



N-Bit SRAM (ff)

- Der Speicher enthält N Speicherzellen L_0, \dots, L_{N-1} , die je ein Bit speichern können.
- Zelle $L_{\langle A \rangle}$ wird mit Hilfe der Adresse A ausgewählt.



N-Bit SRAM (ff)

- Der Speicher enthält N Speicherzellen L_0, \dots, L_{N-1} , die je ein Bit speichern können.
- Zelle $L_{\langle A \rangle}$ wird mit Hilfe der Adresse A ausgewählt.
An D_{out} erscheint der Inhalt von $L_{\langle A \rangle}$.



N-Bit SRAM (ff)

- Der Speicher enthält N Speicherzellen L_0, \dots, L_{N-1} , die je ein Bit speichern können.
- Zelle $L_{\langle A \rangle}$ wird mit Hilfe der Adresse A ausgewählt.
An D_{out} erscheint der Inhalt von $L_{\langle A \rangle}$.
Durch Schreibpuls an W wird D_{in} nach $L_{\langle A \rangle}$ übernommen.

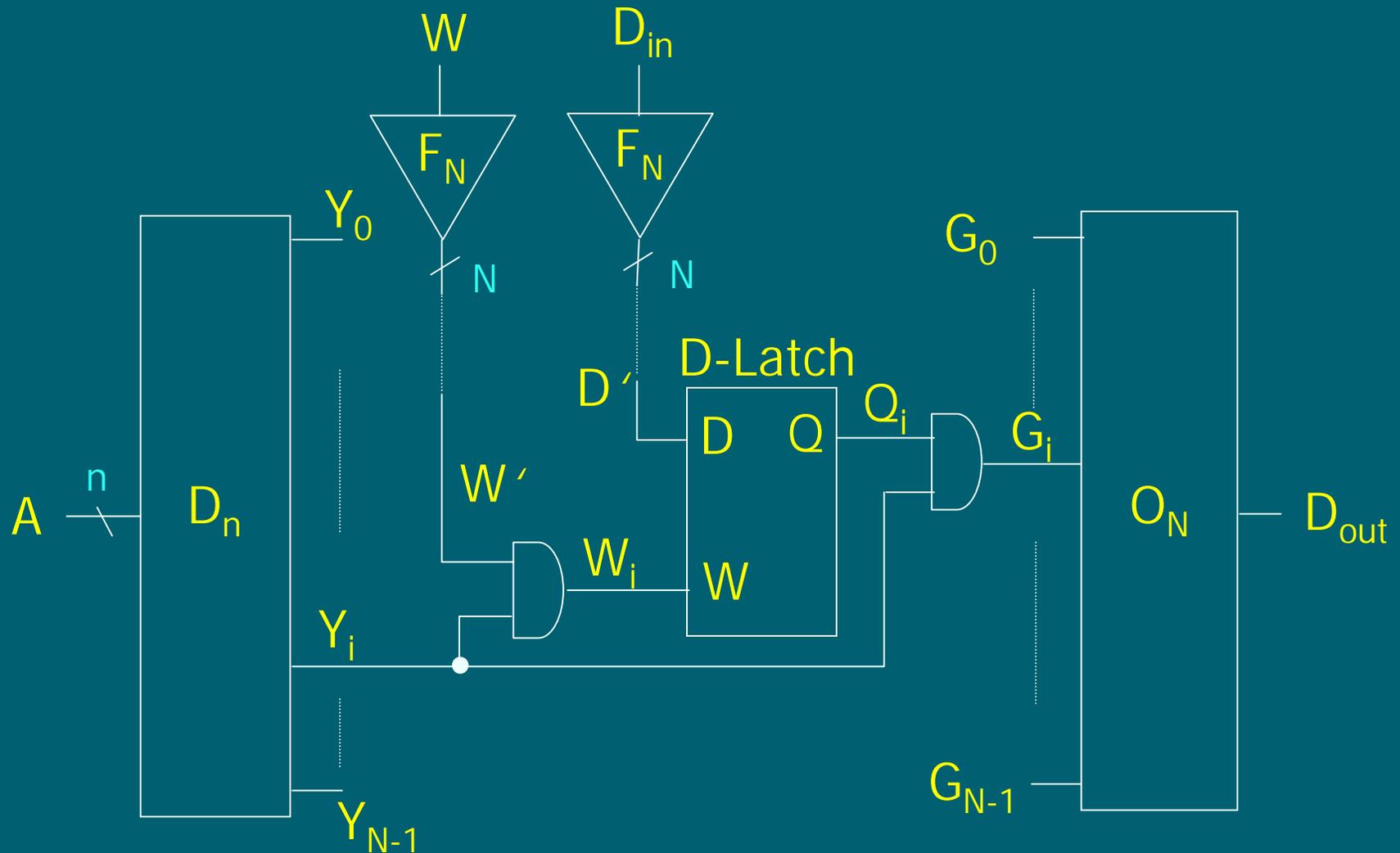


N-Bit SRAM (ff)

- Ein $N \times s$ – SRAM besteht aus s N-Bit SRAMs mit gemeinsamen Adress- und Schreibsignalen.
- s heißt **Bitbreite** des $N \times s$ – SRAMs
- Ein SRAM besteht im Prinzip aus 3 Hilfsschaltkreisen
 1. mehrfaches OR
 2. Treiberbäume
 3. Dekodierer



Schaltbild eines SRAMs: Details werden jetzt entwickelt!



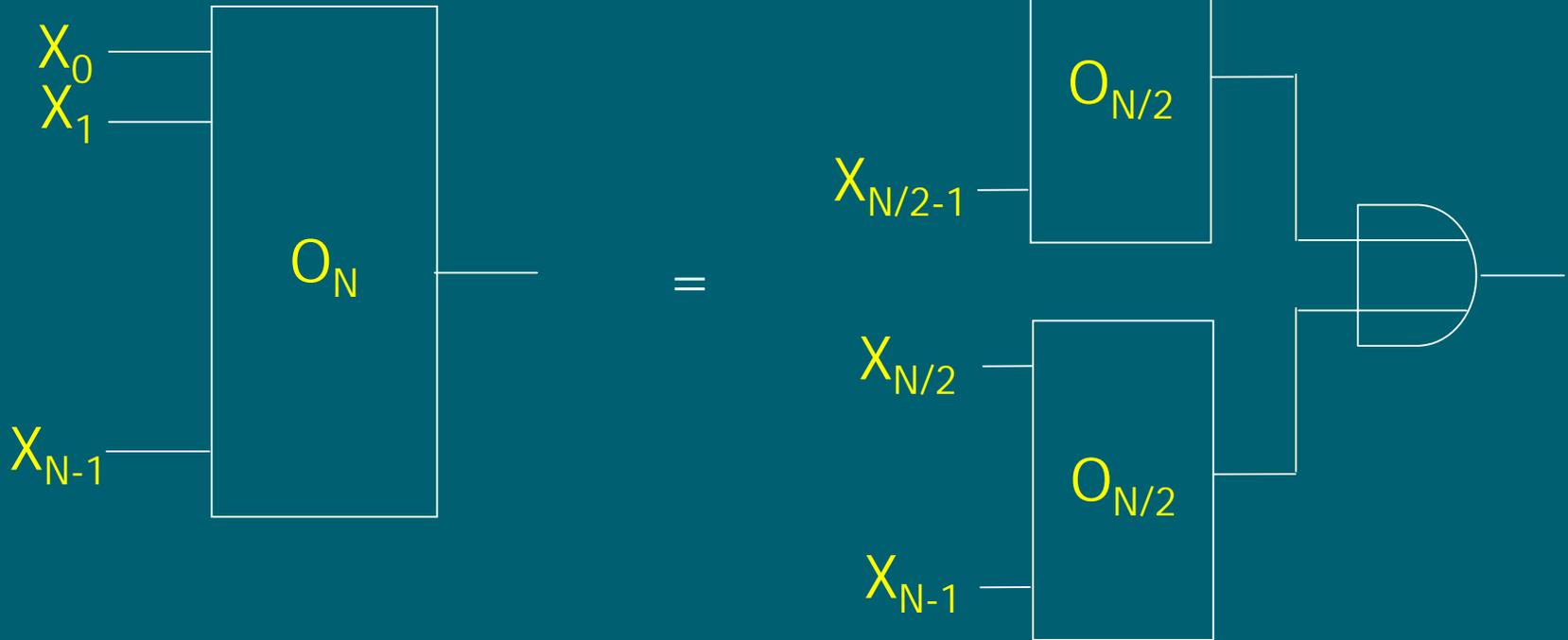
Mehrfaches OR

Ein N-faches OR O_N mit $N = 2^n$ ist ein Schaltkreis, der N-faches Oder berechnet.

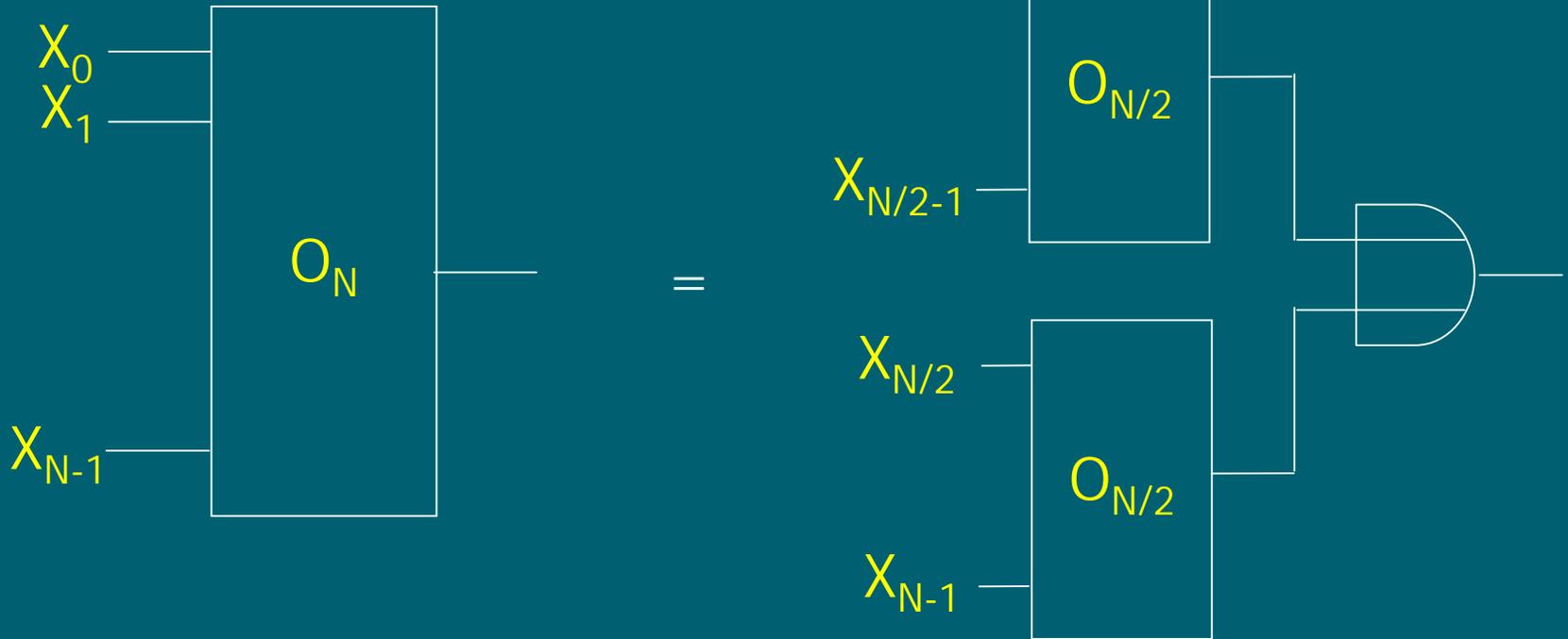


→ balancierter Baum

Mehrfaches OR (ff)

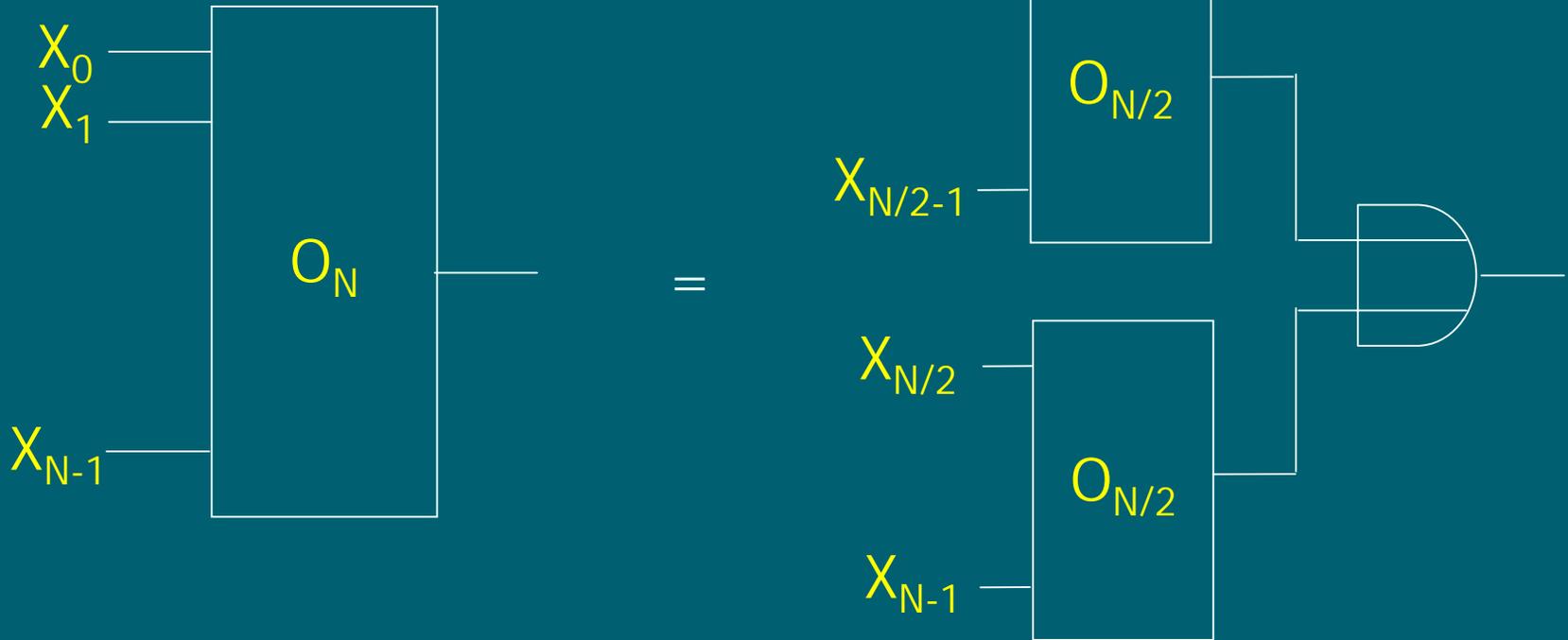


Mehrfaches OR (ff)



	OR	
	min	max
t_{PIH}	3.0	6.6
t_{PHL}	3.0	6.3

Mehrfaches OR (ff)



Verzögerungszeit $\rho(n)$ von O_N :

$$3.0 n \leq \rho(n) \leq 6.6 n$$

	OR	
	min	max
t_{PIH}	3.0	6.6
t_{PHL}	3.0	6.3

Treiberbäume

Ein Treiber ist ein Gatter mit einem Eingang X und einem Ausgang Y , das die Identität $Y = X$ berechnet.

Symbol:



Verzögerungszeiten bei Treibern

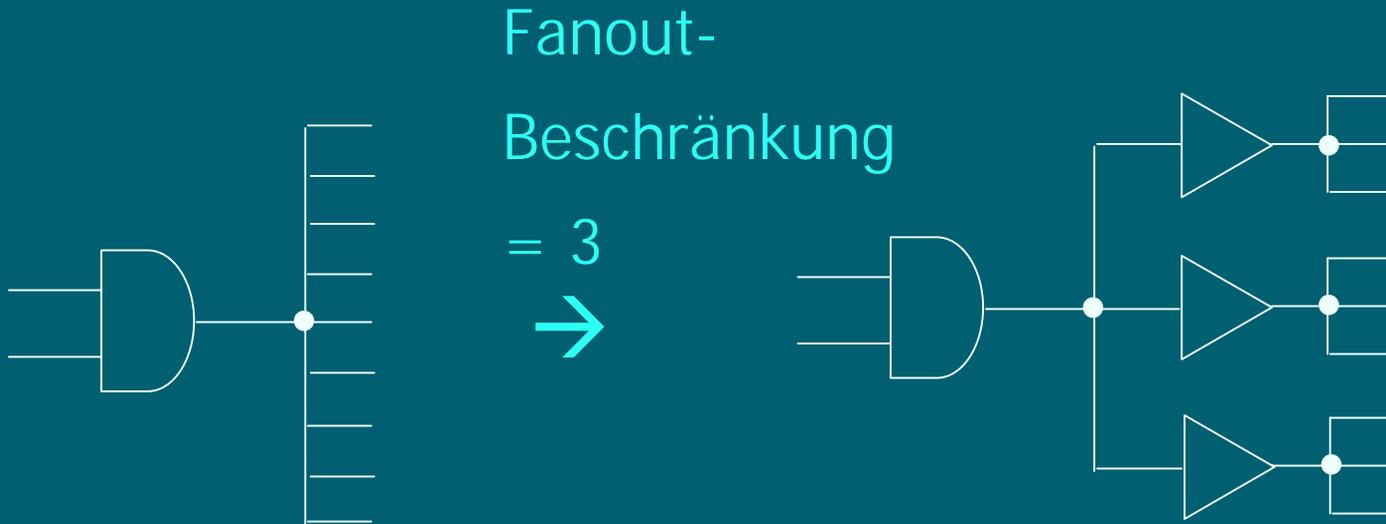
Es gelten folgende Verzögerungszeiten für Treiber (74F244):

	min	max
t_{PLH}	2.5	6.2
t_{PHL}	2.5	6.5

Zweck von Treibern

Zweck: Überwinden von Fanout-Beschränkungen

Beispiel:



Wir betrachten Fanout-Beschränkungen ≤ 10 und konstruieren deshalb Bäume mit Verzweigungsgrad ≤ 10 .

Beachte:

Hier versteht man unter einem Baum
einen (In-)Baum,

Beachte:

Hier versteht man unter einem Baum

einen (In-)Baum,

d.h. einen azyklischen Graphen $G = (V, E)$ mit

Beachte:

Hier versteht man unter einem Baum
einen (In-)Baum,

d.h. einen azyklischen Graphen $G = (V, E)$ mit

- genau einer Quelle w und
 $\text{indeg}(v) = 1$ für alle $v \in V \setminus \{w\}$

Beachte:

Hier versteht man unter einem Baum
einen (In-)Baum,

d.h. einen azyklischen Graphen $G = (V, E)$ mit

- genau einer Quelle w und
 $\text{indeg}(v) = 1$ für alle $v \in V \setminus \{w\}$
- Blätter = Knoten $v \in V$ mit $\text{outdeg}(v) = 0$

Beachte:

Hier versteht man unter einem Baum
einen (In-)Baum,

d.h. einen azyklischen Graphen $G = (V, E)$ mit

- genau einer Quelle w und
 $\text{indeg}(v) = 1$ für alle $v \in V \setminus \{w\}$
- Blätter = Knoten $v \in V$ mit $\text{outdeg}(v) = 0$
- innere Knoten = Knoten $v \in V$ mit $\text{outdeg}(v) \geq 1$

Beachte:

Hier versteht man unter einem Baum
einen (In-)Baum,

d.h. einen azyklischen Graphen $G = (V, E)$ mit

- genau einer Quelle w und
 $\text{indeg}(v) = 1$ für alle $v \in V \setminus \{w\}$
 - Blätter = Knoten $v \in V$ mit $\text{outdeg}(v) = 0$
 - innere Knoten = Knoten $v \in V$ mit $\text{outdeg}(v) \geq 1$
- (Kantenrichtung umgekehrt wie (out-)Baum !!)

Bemerkung

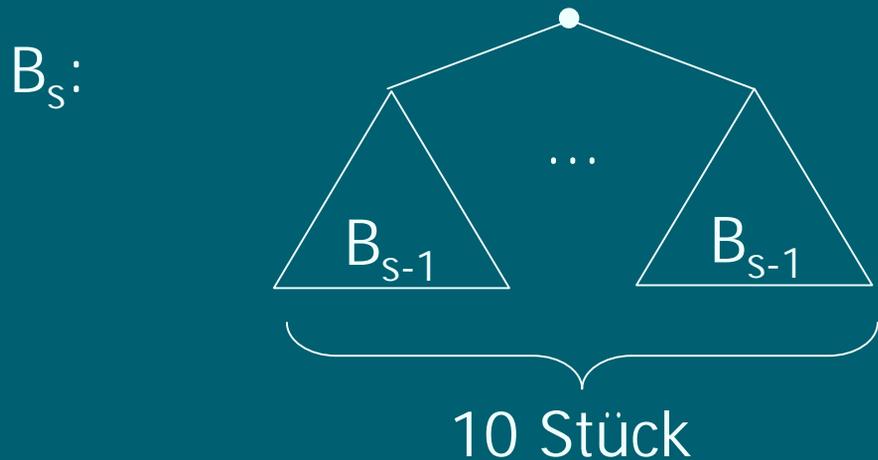
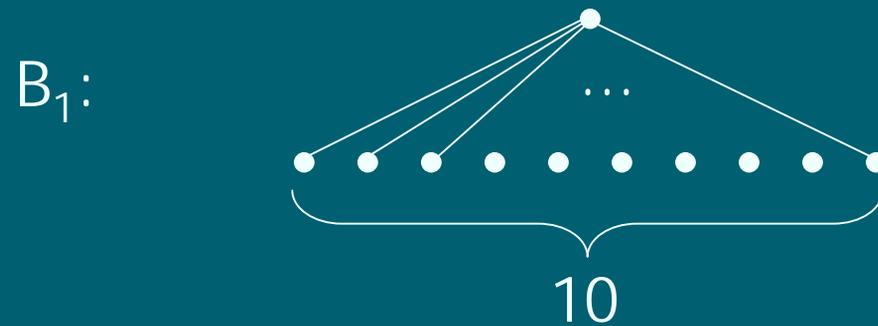
Für Baum B sei

$L(B)$ = Anzahl der Blätter

$I(B)$ = Anzahl innerer Knoten

Betrachte 10-näre Bäume

B_0 : •



Betrachte 10-äre Bäume (ff)

$$L(B_s) = 10^s$$

$$I(B_s) = \sum_{i=0}^{s-1} 10^i = \frac{10^s - 1}{10 - 1} < \frac{L(B_s)}{9}$$

→ Benutze B_s zum x -fachen Vervielfältigen eines Signals mit $x = 10^s$.

Innere Knoten des Baumes werden durch Treiber ersetzt

→ *Treiberbaum* mit Fanoutbeschränkung 10

Mögliches Problem

Ist x mit $10^{s-1} < x < 10^s$ keine Zehnerpotenz,
so konstruiere einen Baum $T(x, s)$, der x Blätter hat
und bei dem alle Pfade von der Wurzel zu einem Blatt
Länge s haben.

Idee:

Fülle Bäume von links her 10-är auf und
sorge zusätzlich für gleiche Tiefe der Blätter!

(Beispiel dazu in der Vorlesung)





Lemma 12.2

$\forall s \in \mathbf{N}$ und $\forall x \in [1, \dots, 10^s]$ gibt es einen Baum $T(x, s)$ mit Ausgangsgrad ≤ 10 an jedem inneren Knoten und den folgenden Eigenschaften:

1. $T(x, s)$ hat x Blätter
2. $I(T(x, s)) \leq \frac{x}{9} + s$
3. Alle Pfade von der Wurzel zu einem Blatt haben Länge s .

Beweis: Induktion über $s \rightarrow$ Übungen

Verzögerungszeit bei Treiberbäumen

- Für $x \in \mathbf{N}$ betrachte $\lambda(x) = \lceil \log_{10} x \rceil$ und konstruiere $T(x, \lambda(x))$.

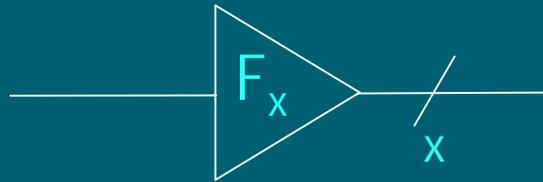
- Wegen $\log_{10} x = \frac{\log x}{\log 10}$ und $3 < \log 10 < 4$

$$\frac{1}{4} \log x < \lambda(x) < \frac{1}{3} \log x + 1 \quad (*)$$

Verzögerungszeit bei Treiberbäumen (ff)

- Ersetze in $T(x, \lambda(x))$ jeden inneren Knoten durch Treiber
→ Schaltkreis F_x mit 1 Eingang und x Ausgängen.

Symbol:



→ Verzögerungszeit $\tau(x)$ von F_x :

$$2.5 \lambda(x) \leq \tau(x) \leq 6.5 \lambda(x)$$

und wegen (*)

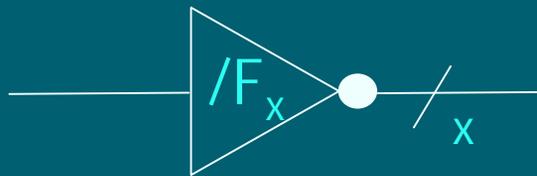
$$0.625 \log x \leq \tau(x) < 2.2 \log x + 6.5$$

	min	max
t_{PLH}	2.5	6.2
t_{PHL}	2.5	6.5

Verzögerungszeit bei Treiberbäumen (ff)

- Invertierender Treiberbaum $/F_x$:
Ersetze in F_x Treiber an der Wurzel durch Inverter
→ Gegenüber F_x
minimale Verzögerungszeit um 1 ns geringer,
maximale Verzögerungszeit um 0.5 ns.

Symbol:



Definition

Sei $n \in \mathbf{N}$, $N = 2^n$.

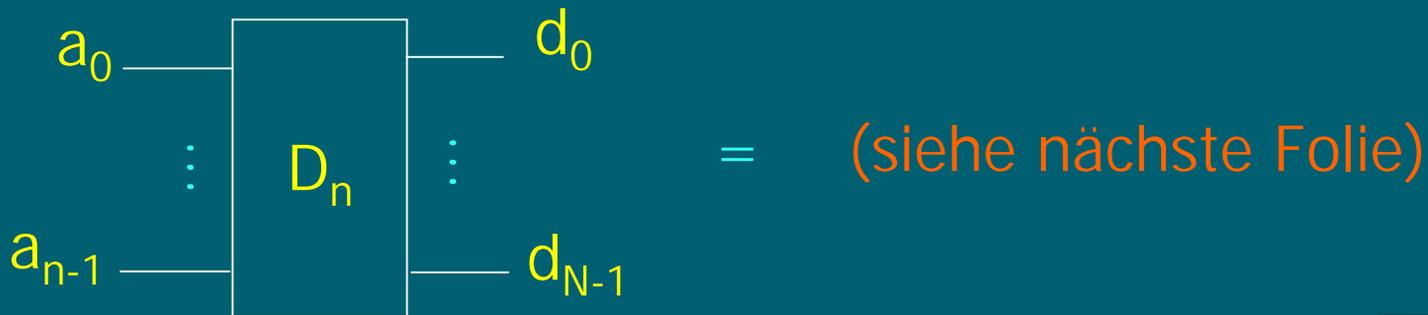
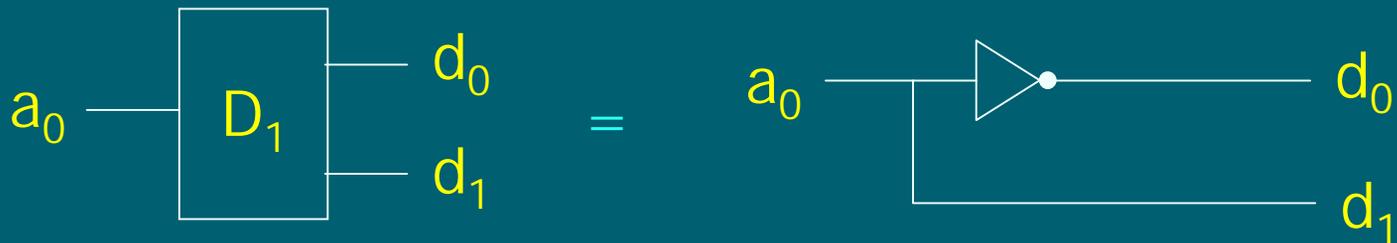
Ein **n-Bit-Dekodierer** D_n ist ein Schaltkreis, der die Funktion $d: \mathbf{B}^n \rightarrow \mathbf{B}^N$ berechnet, wobei gilt:

$$d_i(a) = \begin{cases} 1, & \text{falls } \langle a \rangle = i \\ 0, & \text{sonst} \end{cases} \quad \forall i = 0, \dots, N-1$$

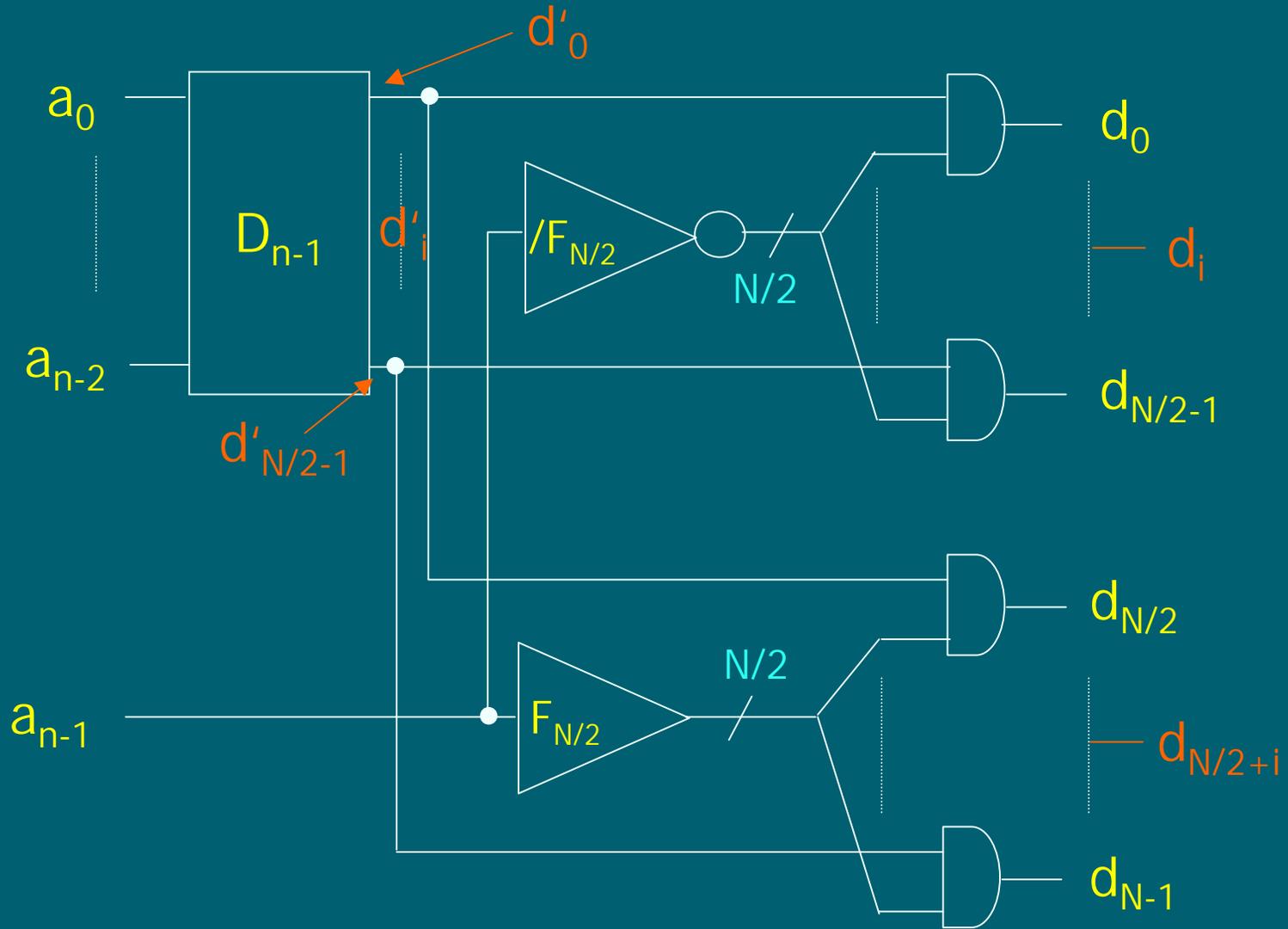
($d_i(a)$ ist Bit i des N -Tupels $d(a)$.)

Dekodierer - Konstruktionsidee

Baue D_n rekursiv aus D_{n-1} und Zusatzlogik.



n-Bit-Dekodierer



Verzögerungszeit beim Dekodierer

Obere Abschätzung für Verzögerungszeit $\delta(n)$ von D_n :

$$\delta(1) \leq 6.0,$$

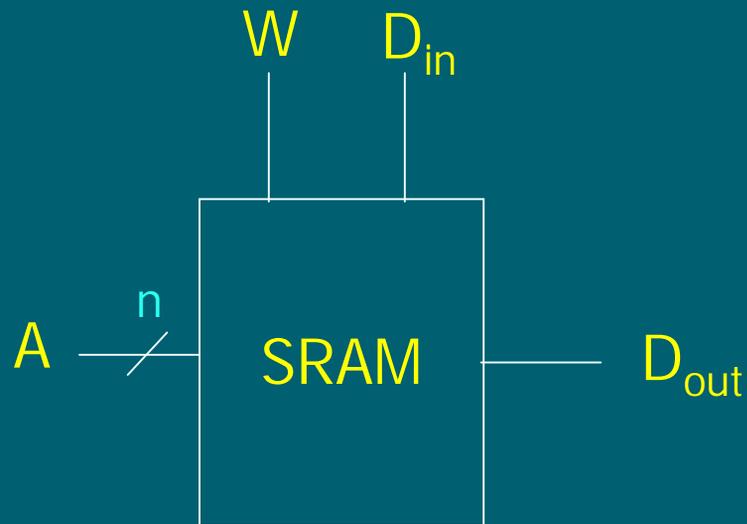
$$\delta(2) \leq 6.6 + \max\{ 6.0, \tau(2) \} \leq 2 \cdot 6.6$$

$$\begin{aligned} \delta(n) &\leq 6.6 + \max\{ \delta(n-1), \tau(2^{n-1}) \} \\ &\leq 6.6 + \underbrace{\max\{ \delta(n-1), 2.2(n-1) + 6.5 \}} \\ &\leq 6.6(n-1) \\ &\text{nach I.V.} \end{aligned}$$

Durch Induktion folgt:

$$\delta(n) \leq 6.6 n$$

Aufbau eines SRAM



Das SRAM besteht aus $N = 2^n$ D-Latches L_0, \dots, L_{N-1} .

Schreib- und Lese-Vorgang

Lesen:

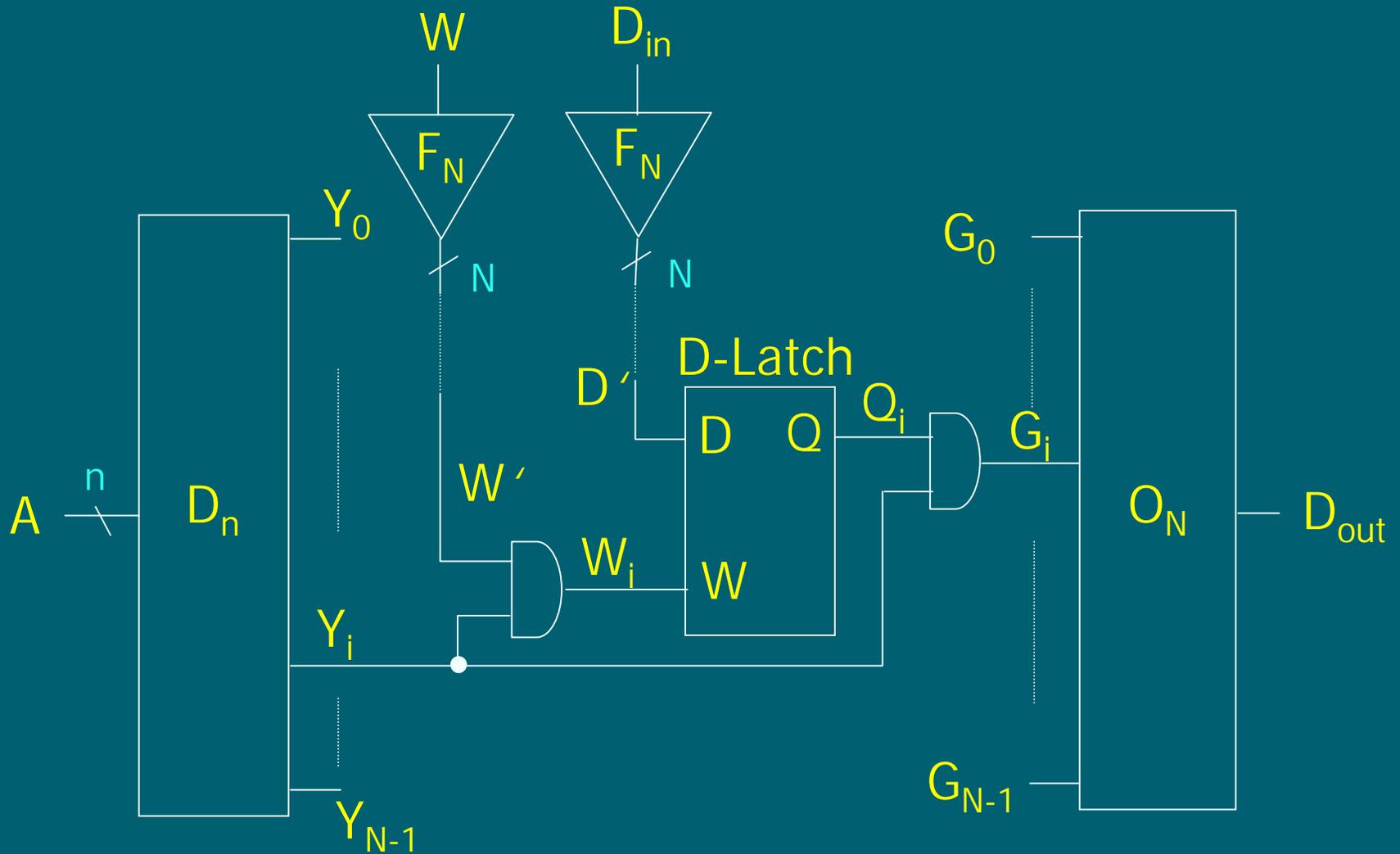
An D_{out} soll der Inhalt von $L_{\langle A \rangle}$ ausgegeben werden.

Schreiben:

Bei Schreibpuls W soll D_{in} nach $L_{\langle A \rangle}$ übernommen werden.



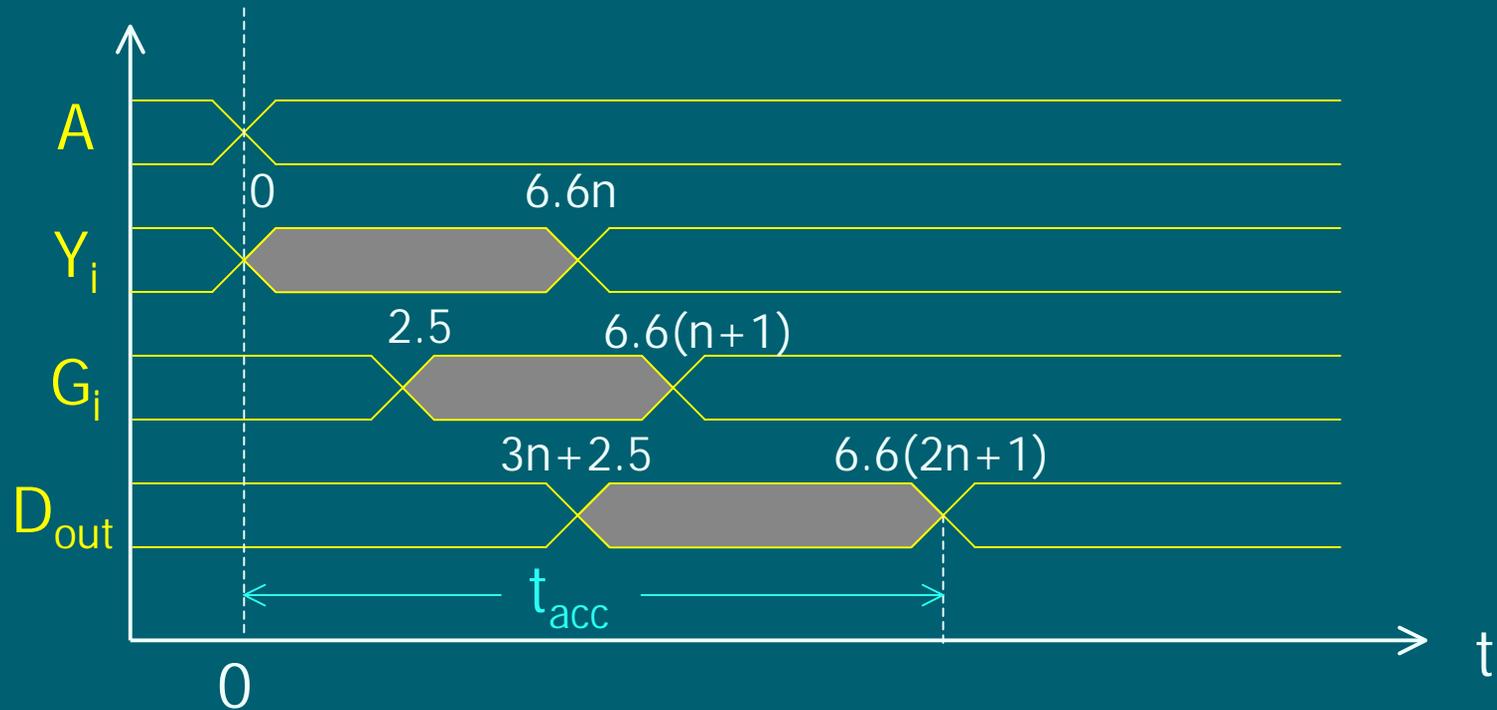
Schaltbild eines SRAMs



Bemerkungen zur Schaltung

- Die vorige Schaltung realisiert ein SRAM funktionell.
- Timing-Analyse nötig, um zu klären, wie lange jeweilige Signale anliegen müssen für zeitlich korrektes Verhalten.

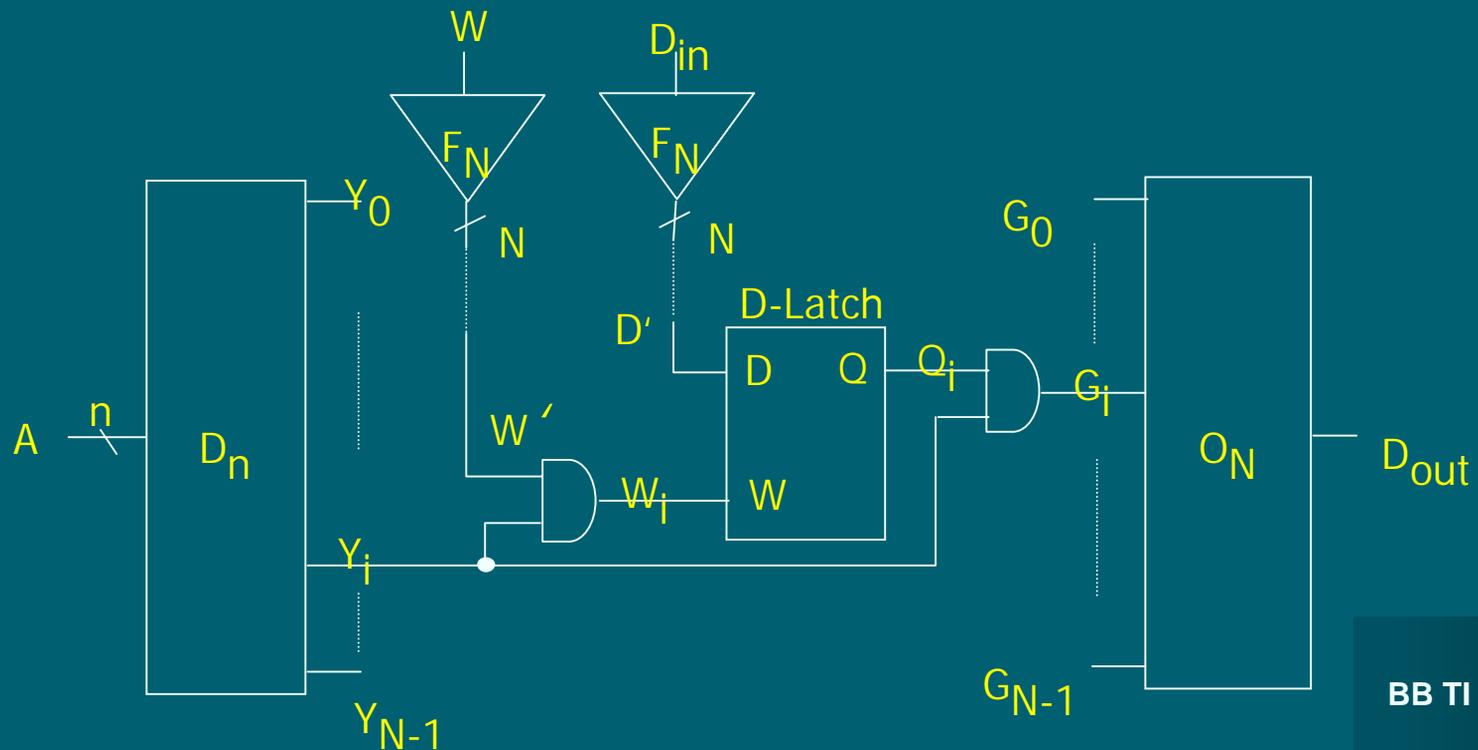
Timing-Analyse für Lesezugriff



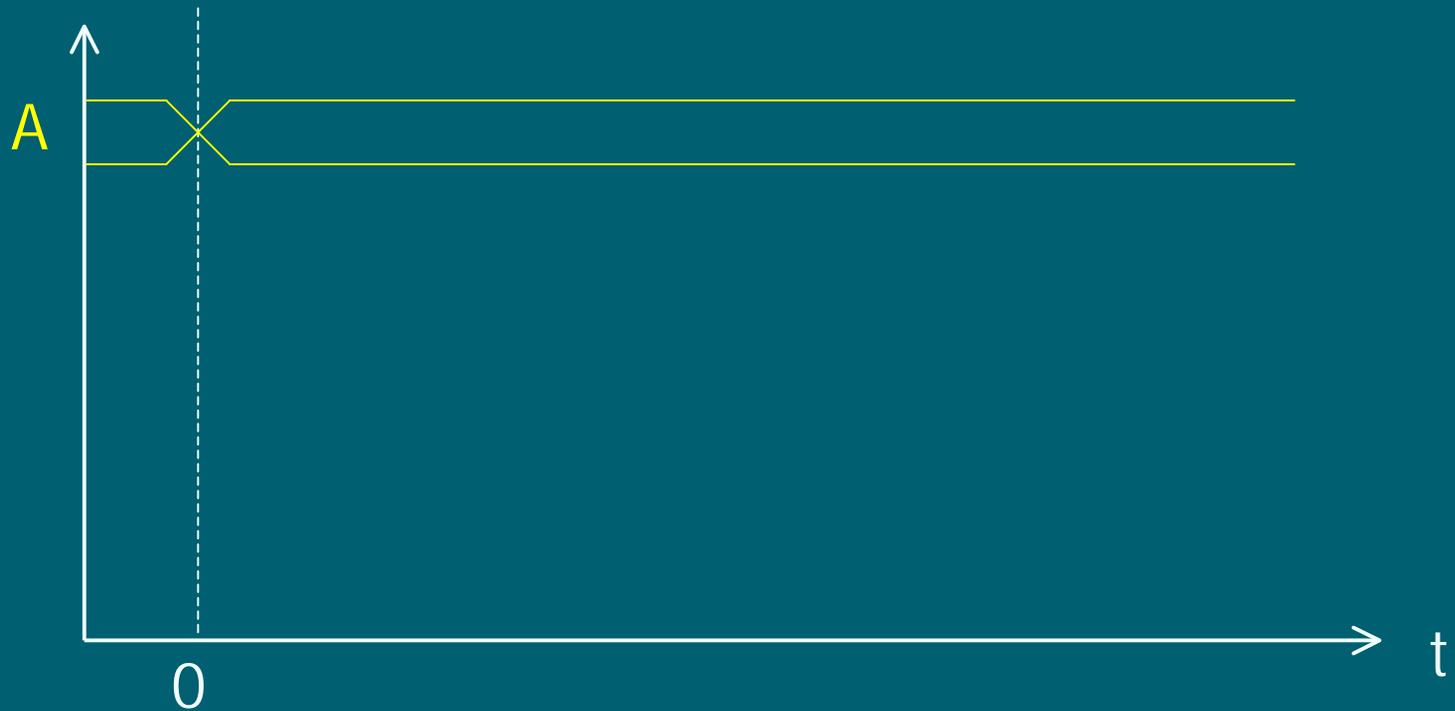
Timing-Analyse für Lesezugriff (ff)

Adresse stabil bei $t = 0$.

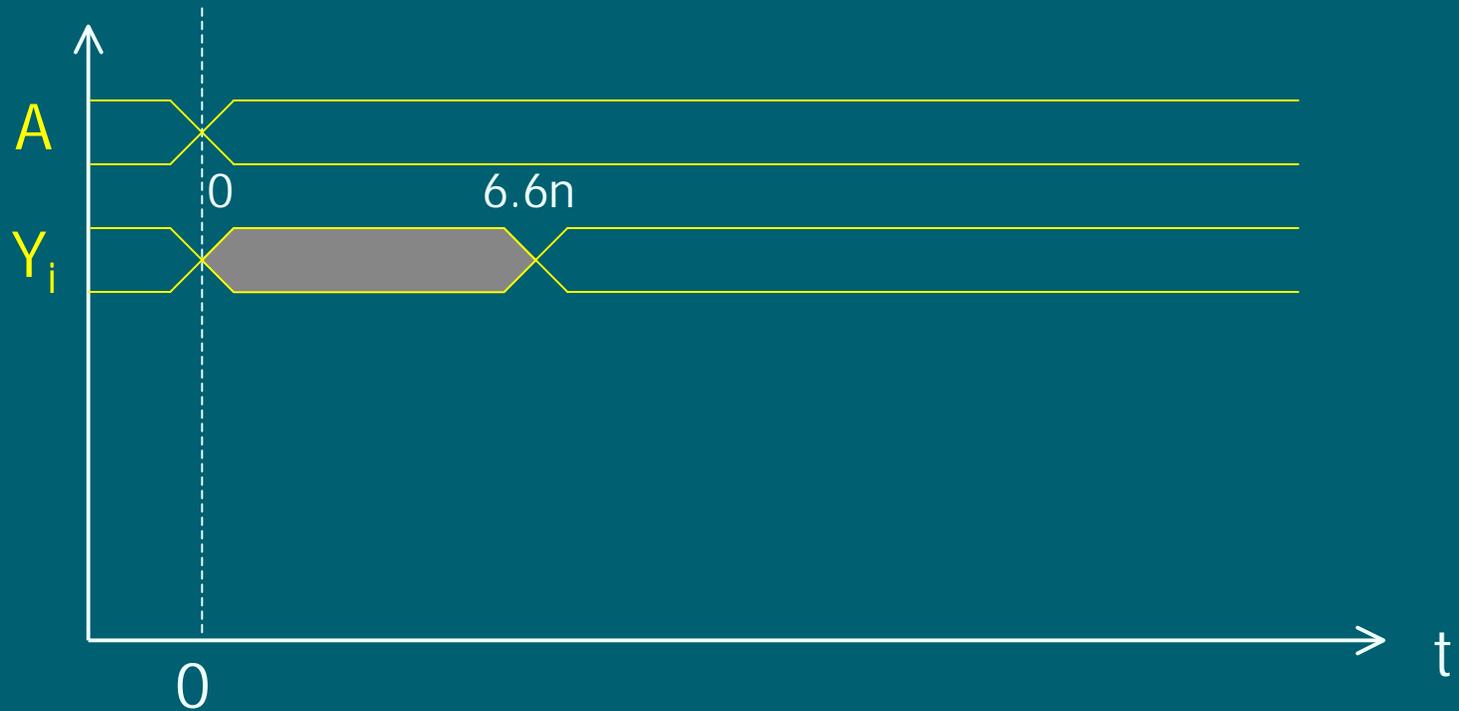
→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder



Timing-Analyse für Lesezugriff



Timing-Analyse für Lesezugriff

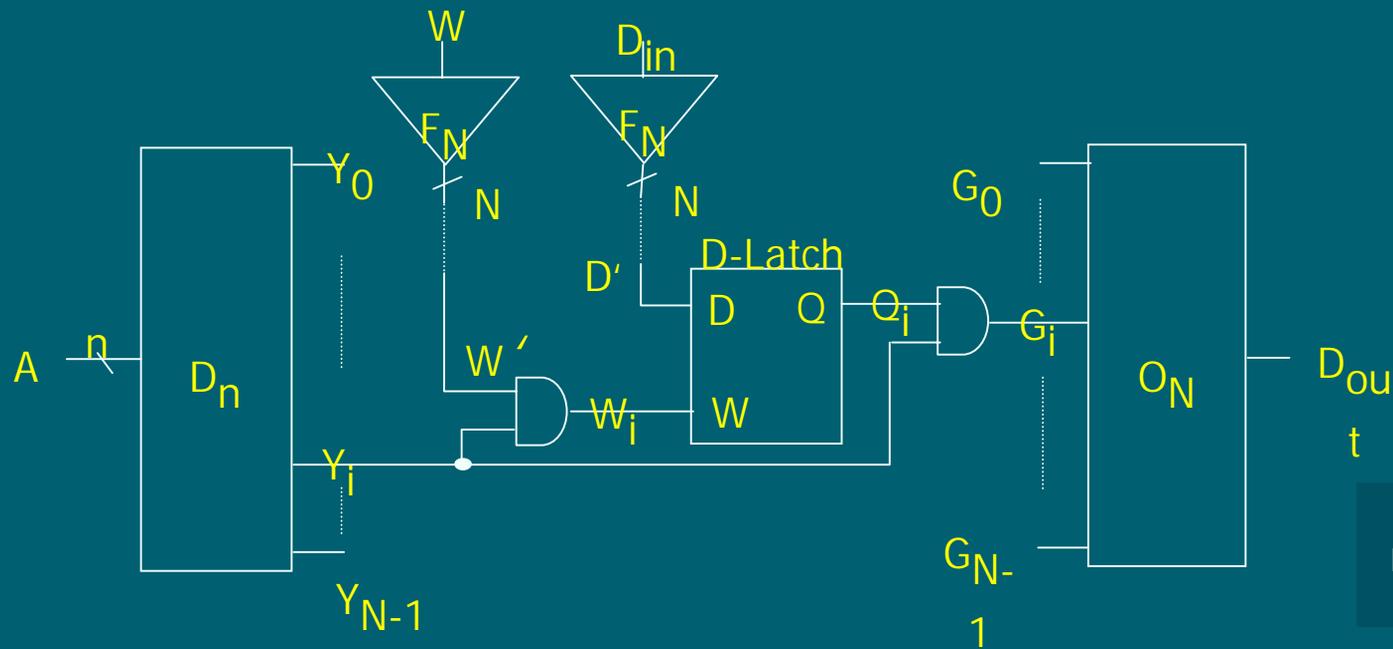


Timing-Analyse für Lesezugriff (ff)

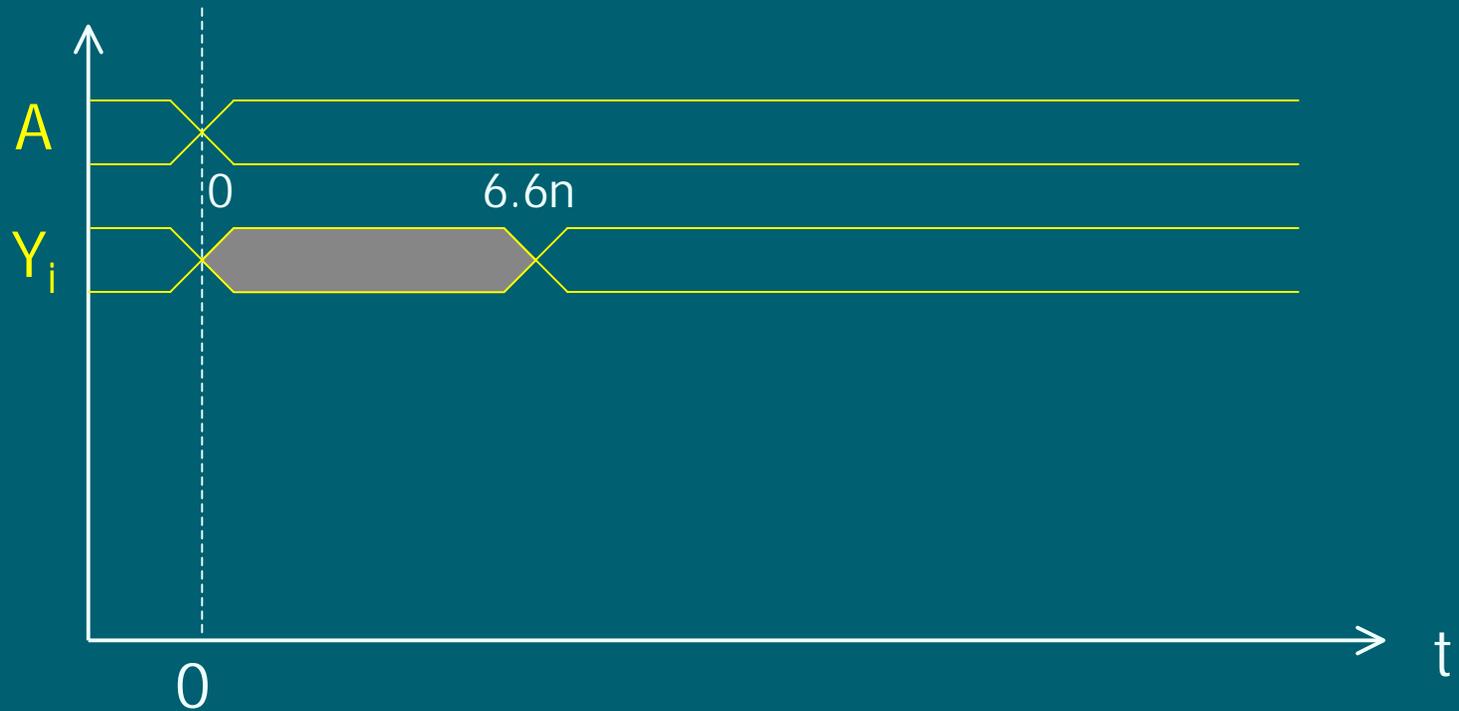
Adresse stabil bei $t = 0$.

→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder

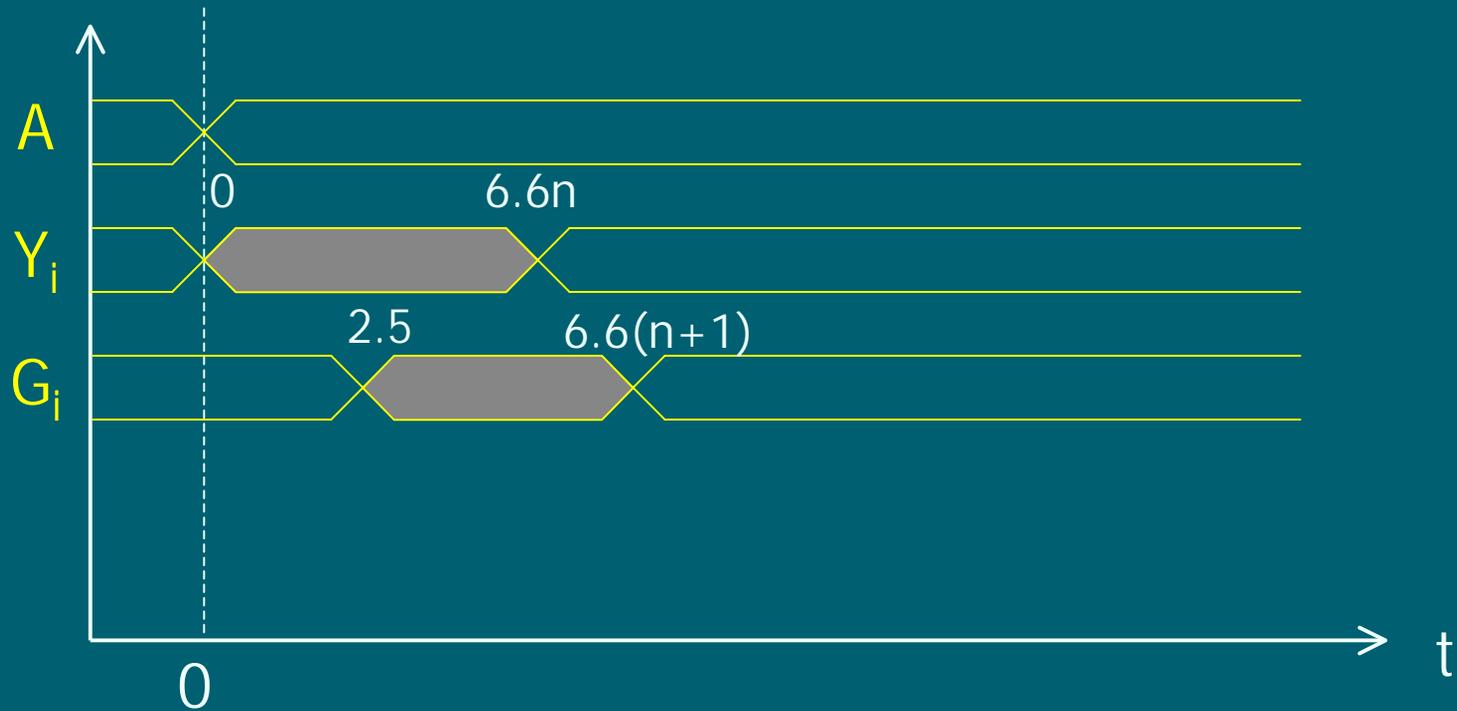
→ G_i stabil zur Zeit $t_2 = t_1 + (2.5, 6.6)$ ← AND
 $= (2.5, 6.6(n+1))$



Timing-Analyse für Lesezugriff



Timing-Analyse für Lesezugriff



Timing-Analyse für Lesezugriff (ff)

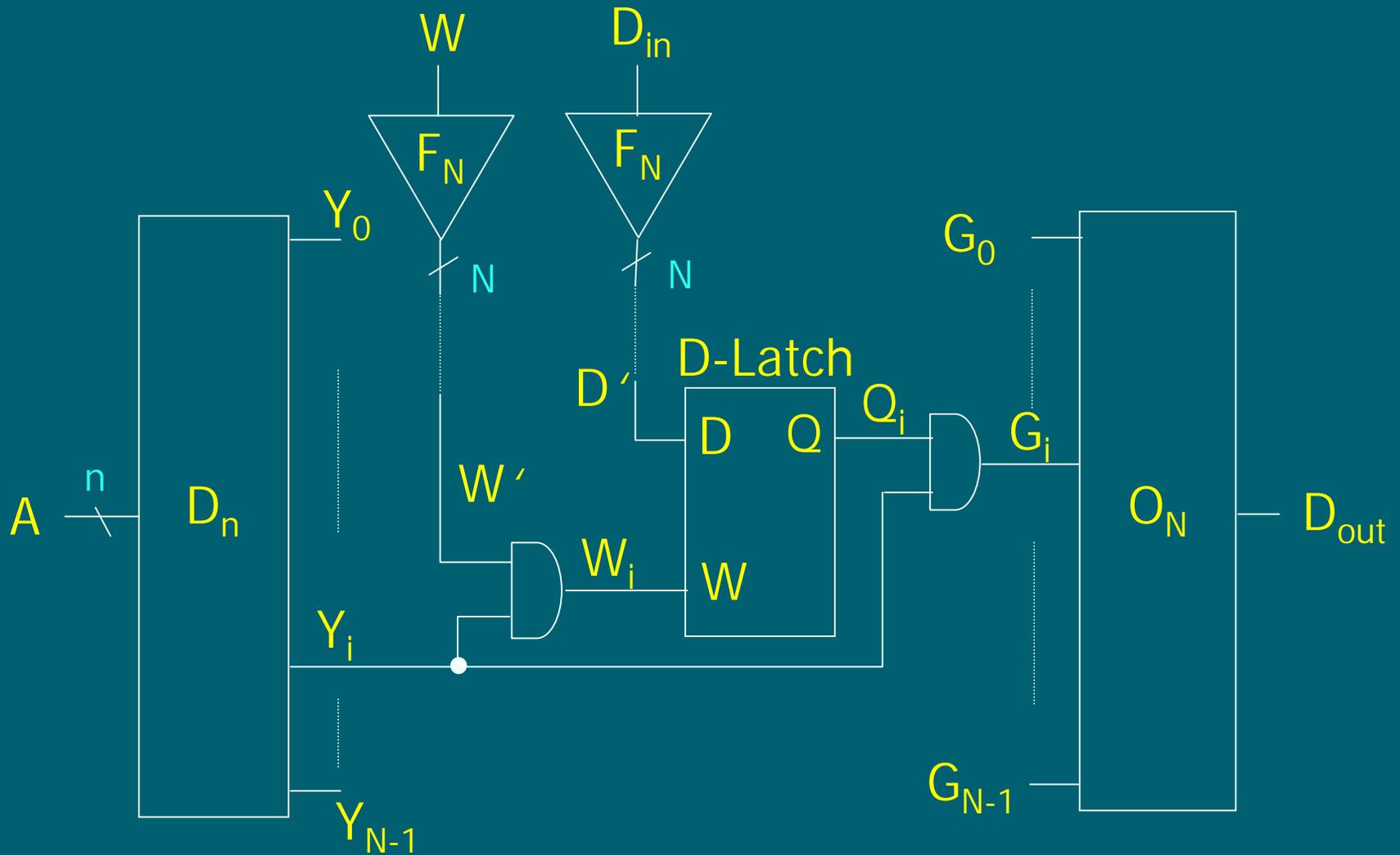
Adresse stabil bei $t = 0$.

→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder

→ G_i stabil zur Zeit $t_2 = t_1 + (2.5, 6.6)$ ← AND
= $(2.5, 6.6(n+1))$

→ D_{out} stabil zur Zeit t_{acc} ??

Schaltbild eines SRAMs



Timing-Analyse für Lesezugriff (ff)

Adresse stabil bei $t = 0$.

→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder

→ G_i stabil zur Zeit $t_2 = t_1 + (2.5, 6.6)$ ← AND
= $(2.5, 6.6(n+1))$

Timing-Analyse für Lesezugriff (ff)

Adresse stabil bei $t = 0$.

→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder

→ G_i stabil zur Zeit $t_2 = t_1 + (2.5, 6.6)$ ← AND
 $= (2.5, 6.6(n+1))$

→ D_{out} stabil zur Zeit $t_{acc} = t_2 + \rho(n)$ ← OR-Baum

Timing-Analyse für Lesezugriff (ff)

Adresse stabil bei $t = 0$.

→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder

→ G_i stabil zur Zeit $t_2 = t_1 + (2.5, 6.6)$ ← AND
 $= (2.5, 6.6(n+1))$

→ D_{out} stabil zur Zeit $t_{acc} = t_2 + \rho(n)$ ← OR-Baum
 $= (3n+2.5, 6.6(2n+1))$

Timing-Analyse für Lesezugriff (ff)

Adresse stabil bei $t = 0$.

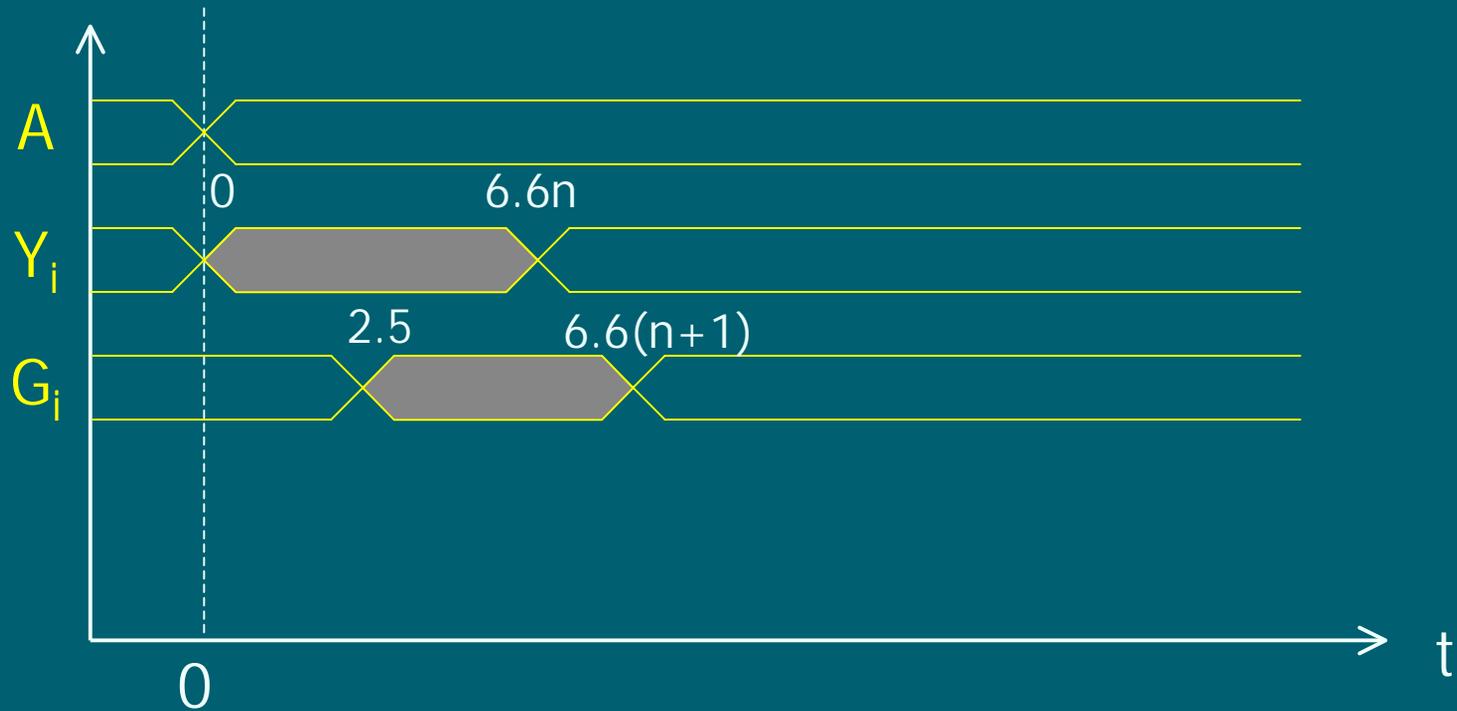
→ Y_i stabil zur Zeit $t_1 = \delta(n) = (0, 6.6n)$ ← Decoder

→ G_i stabil zur Zeit $t_2 = t_1 + (2.5, 6.6)$ ← AND
 $= (2.5, 6.6(n+1))$

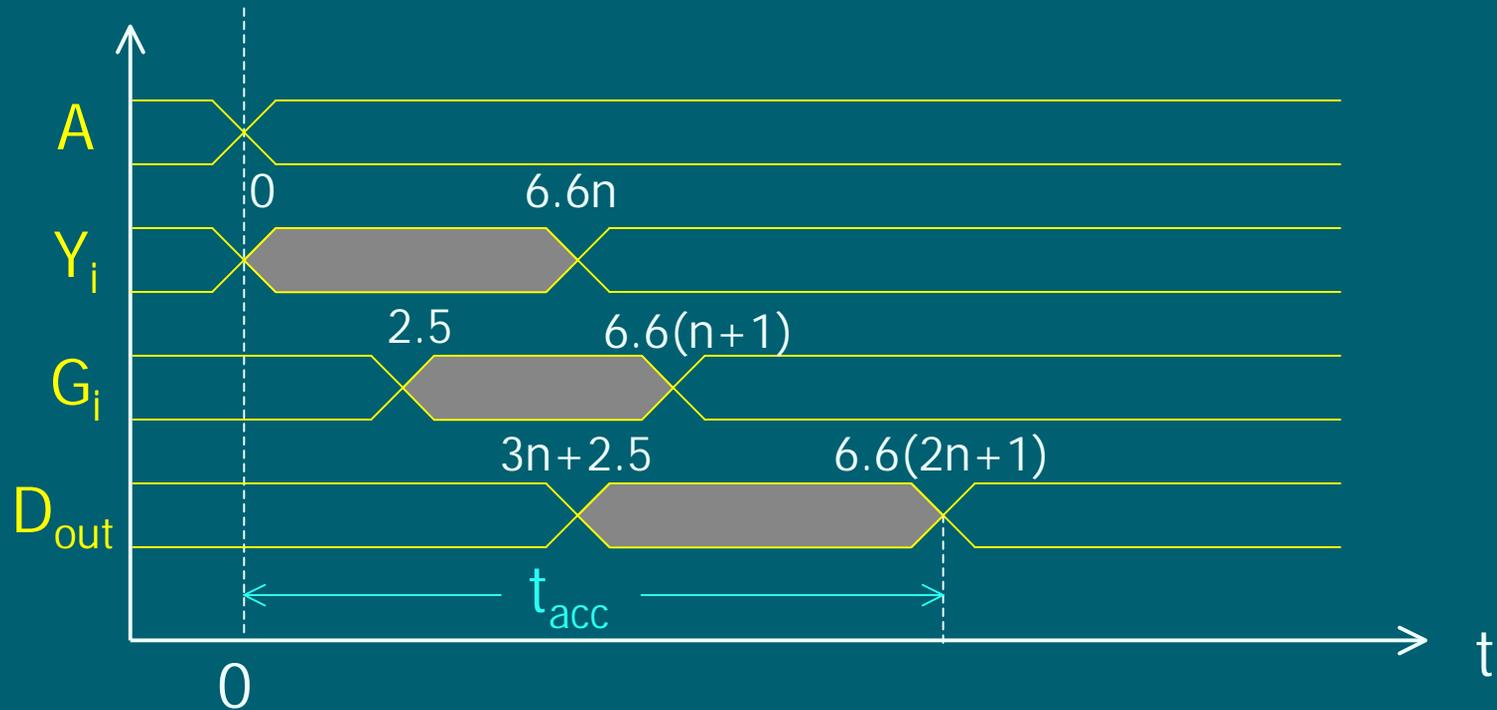
→ D_{out} stabil zur Zeit $t_{acc} = t_2 + \rho(n)$ ← OR-Baum
 $= (3n+2.5, 6.6(2n+1))$

$t_{acc} =$ Lesezugriffszeit

Timing-Analyse für Lesezugriff



Timing-Analyse für Lesezugriff

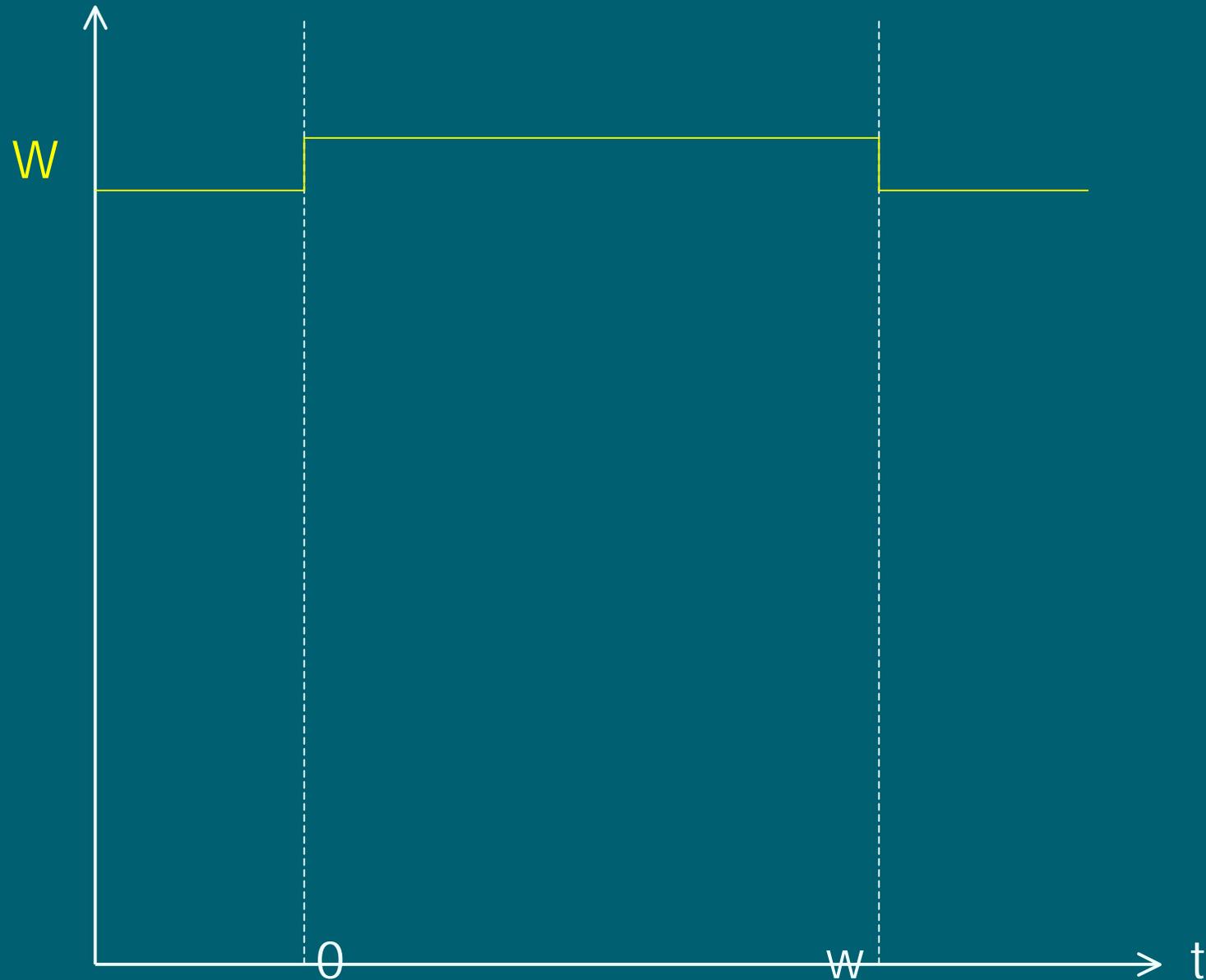


Timing-Analyse für Schreibzugriff

Der Schreibzugriff basiert auf folgenden Parametern:

- Das Schreibsignal W wird von 0 bis w aktiviert

Timing-Diagramm

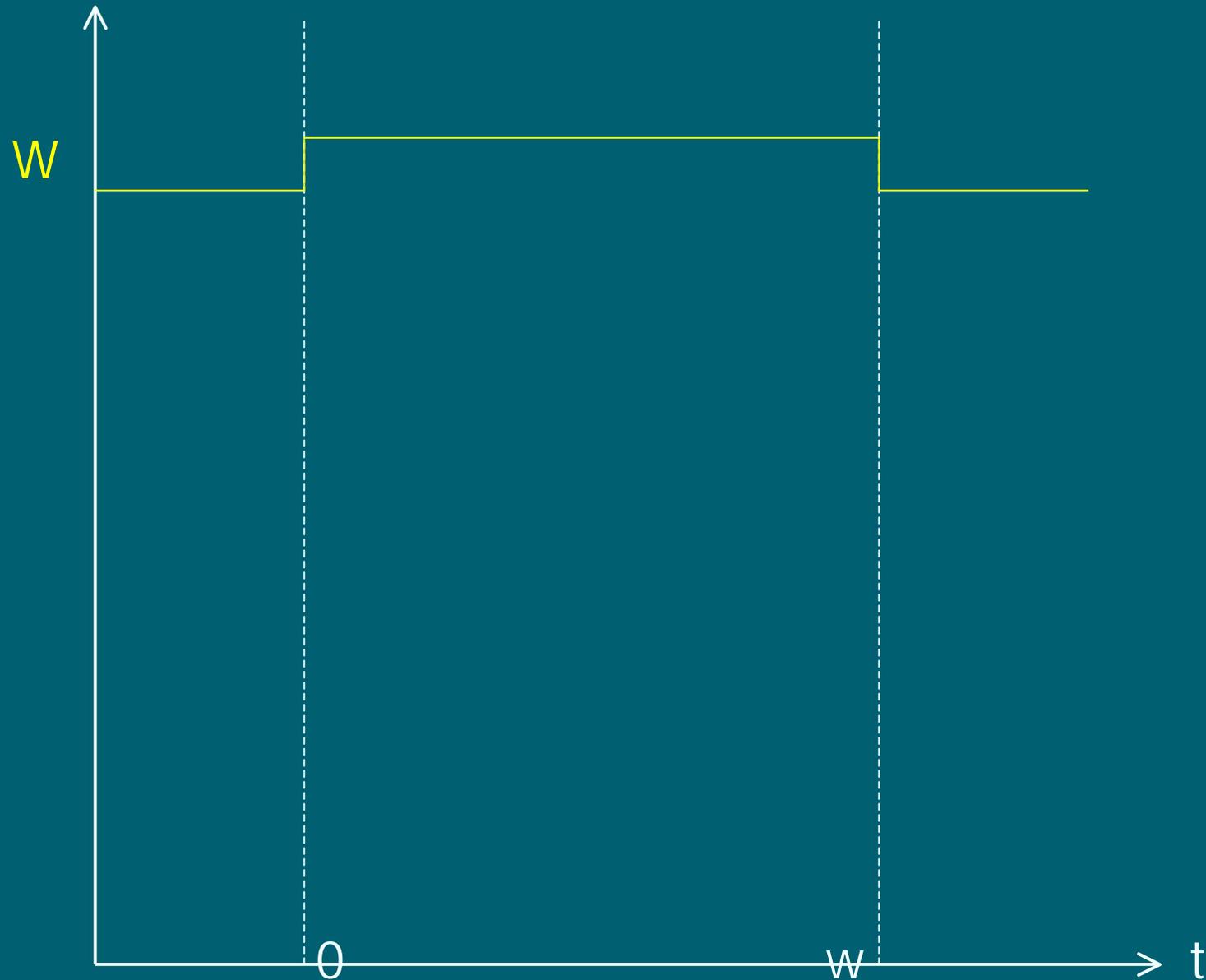


Timing-Analyse für Schreibzugriff

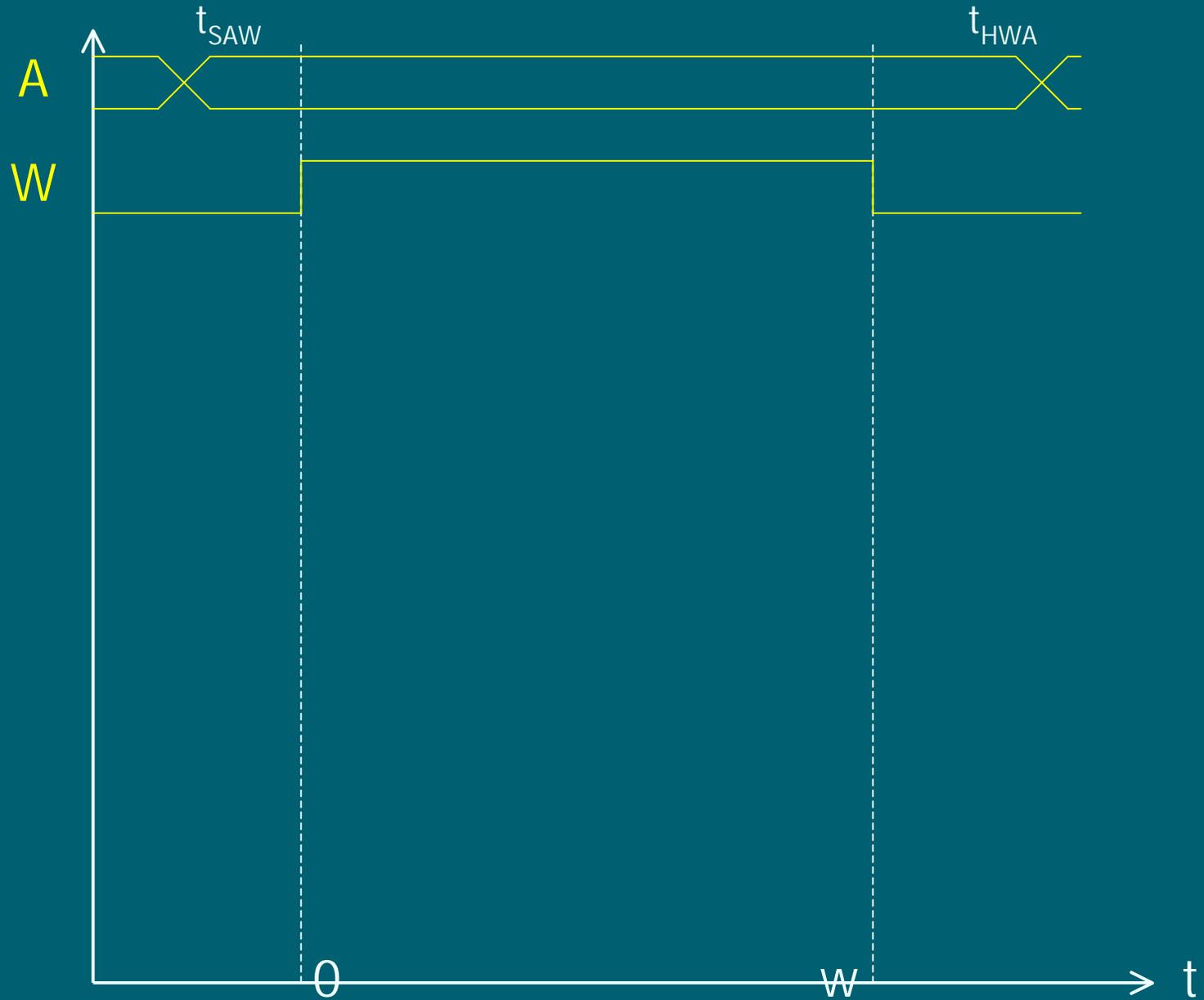
Der Schreibzugriff basiert auf folgenden Parametern:

- Das Schreibsignal W wird von 0 bis w aktiviert
- Adressen werden stabil gehalten von $-t_{SAW}$ bis Zeit $w + t_{HWA}$

Timing-Diagramm



Timing-Diagramm

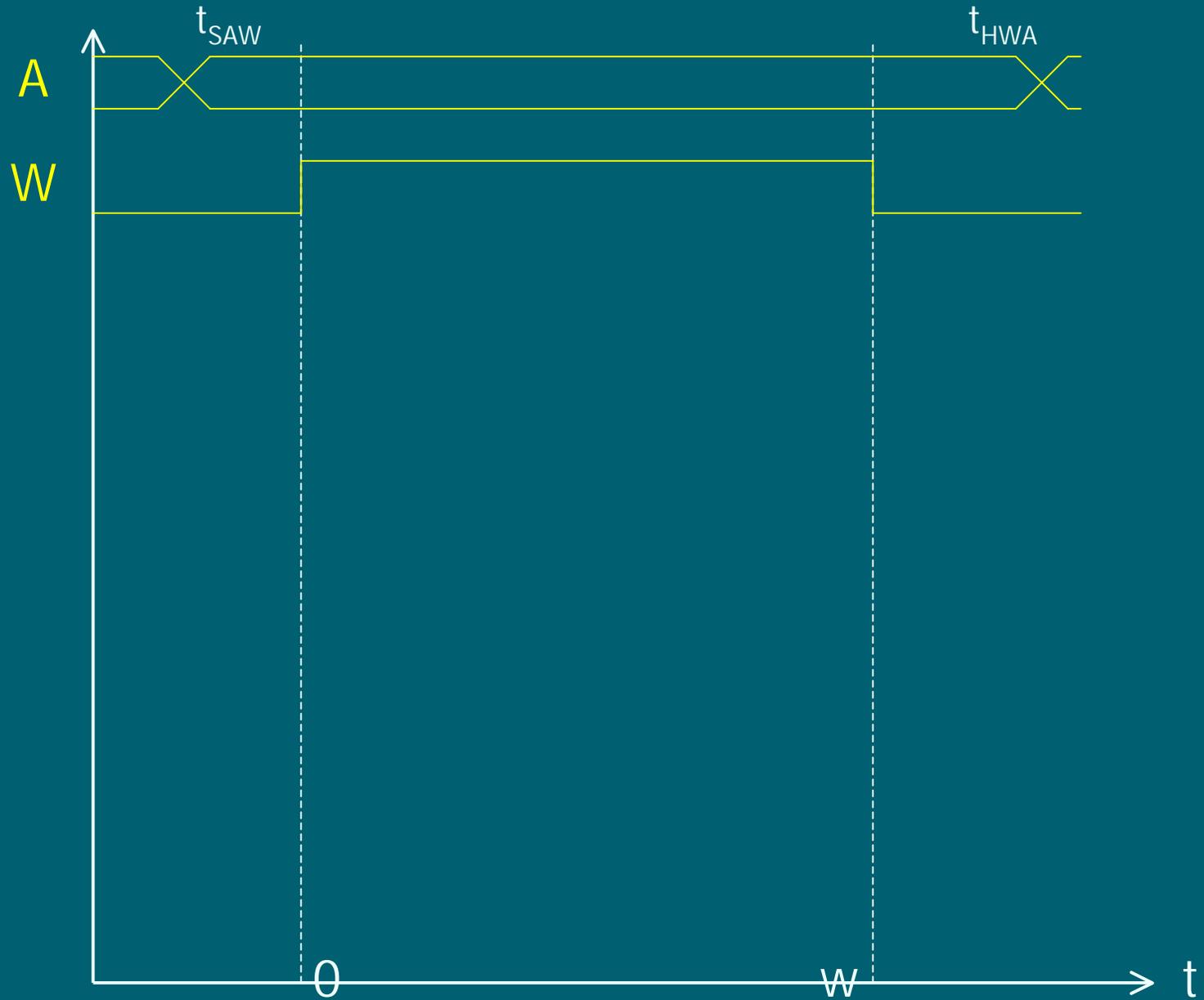


Timing-Analyse für Schreibzugriff

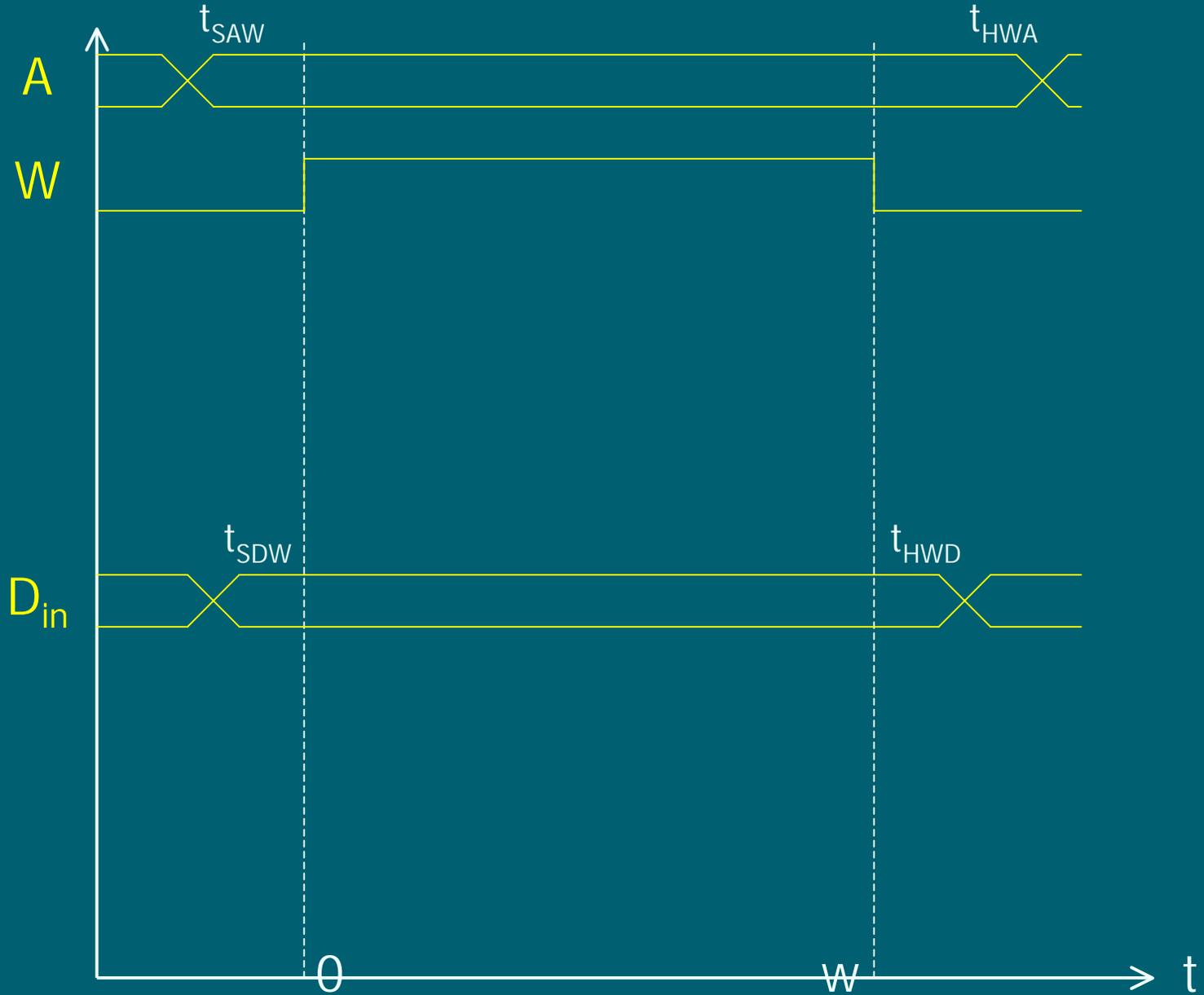
Der Schreibzugriff basiert auf folgenden Parametern:

- Das Schreibsignal W wird von 0 bis w aktiviert
- Adressen werden stabil gehalten von $-t_{SAW}$ bis Zeit $w + t_{HWA}$
- D_{in} wird stabil gehalten von $-t_{SDW}$ bis $w + t_{HWD}$

Timing-Diagramm



Timing-Diagramm

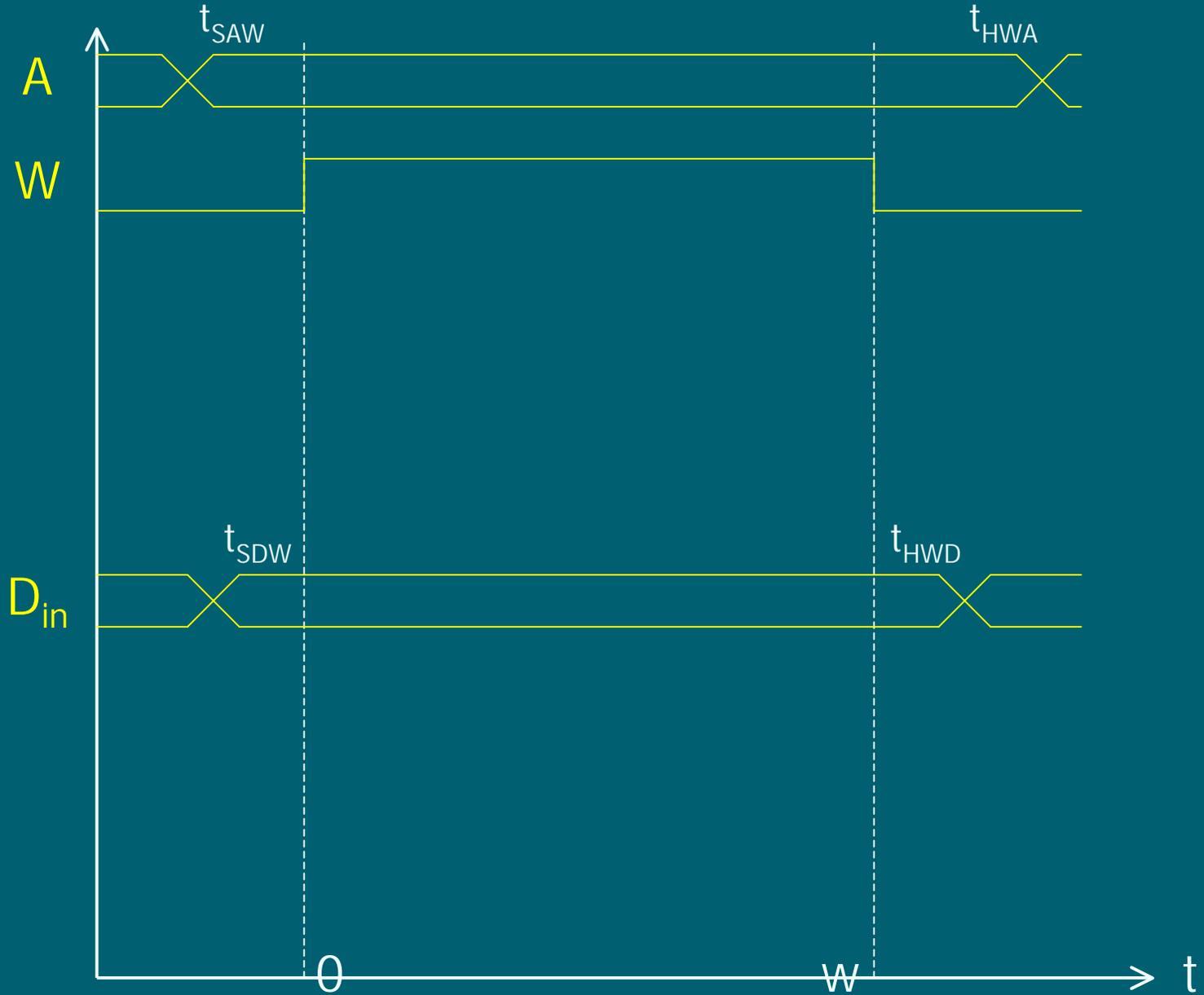


Timing-Analyse für Schreibzugriff

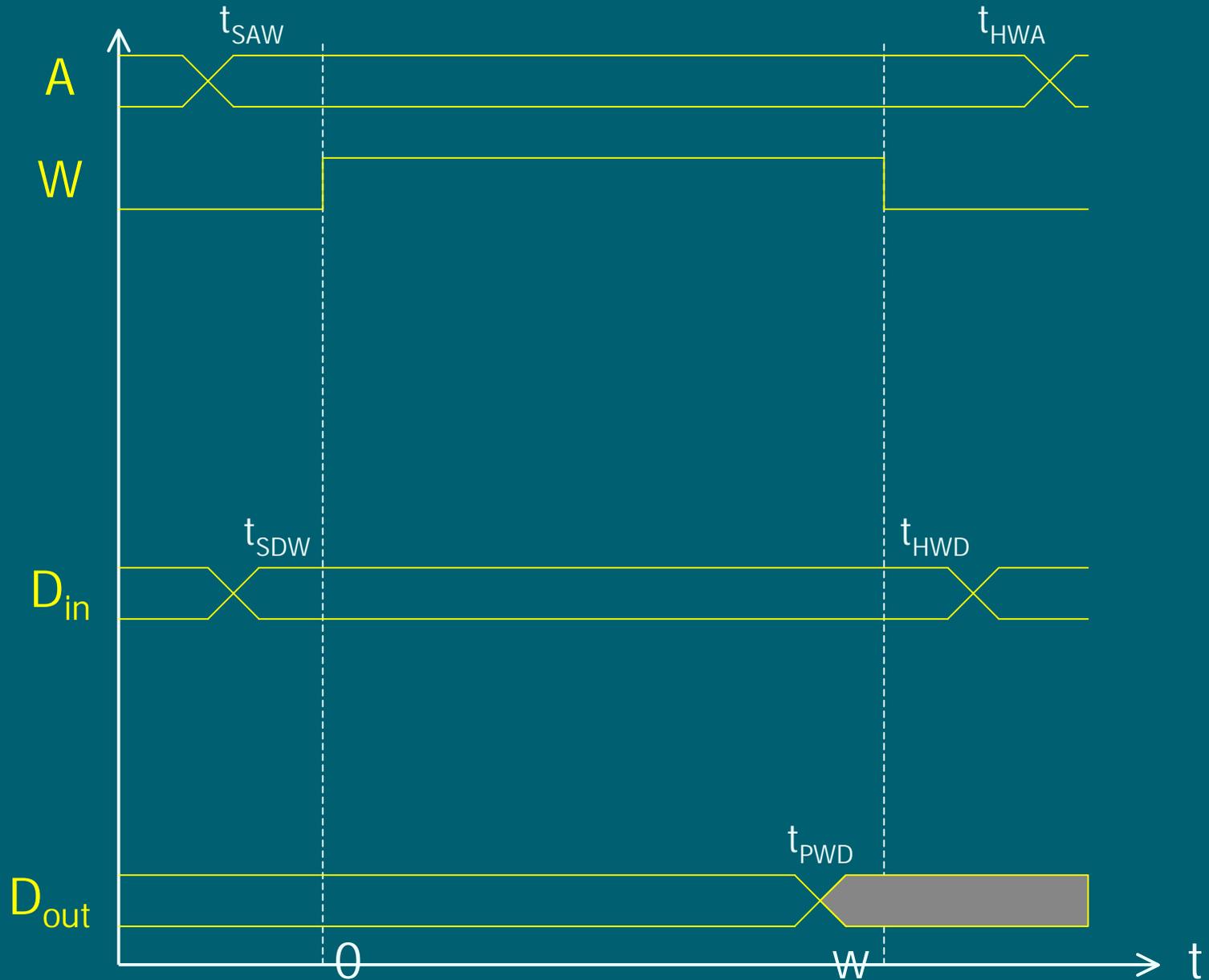
Der Schreibzugriff basiert auf folgenden Parametern:

- Das Schreibsignal W wird von 0 bis w aktiviert
- Adressen werden stabil gehalten von $-t_{SAW}$ bis Zeit $w + t_{HWA}$
- D_{in} wird stabil gehalten von $-t_{SDW}$ bis $w + t_{HWD}$
- Die neuen Daten sind zur Zeit t_{PWD} an D_{out} beobachtbar

Timing-Diagramm



Timing-Diagramm



Symbole und Bezeichnungen

Symbol	Bezeichnung	min	max	
W	Schreibpulsweite	$1.575n + 35.8$		I.
t_{SAW}	Setup-Zeit von A bis W	$5.975n + 11.3$		II.
t_{HWA}	Hold-Zeit von A nach W	$2.2n + 17.8$		II.
t_{SDW}	Setup-Zeit von D bis W	$1.575n + 19.8$		III.
t_{HWD}	Hold-Zeit von D nach W	$1.575n + 23.8$		IV.
t_{PWD}	Verzögerungszeit von W bis D	$3.625n + 9.4$	$8.8n + 36.6$	

Korrektheit

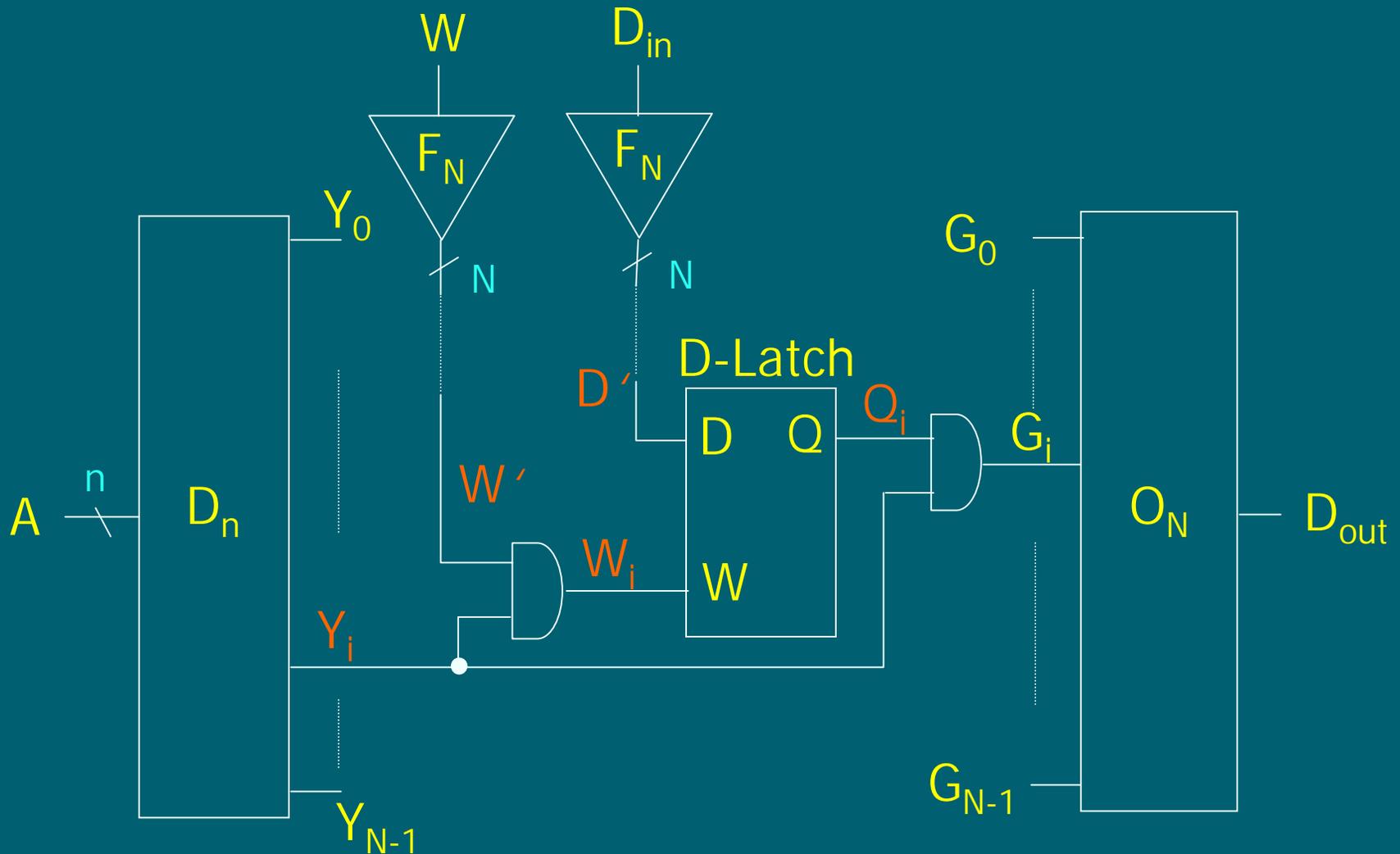
Man kann zeigen, dass der Schreibzyklus mit den gegebenen Parameterwerten gelingt.

Beweis: → Übungen

Analoge Zeiten werden für den Betrieb von anderen SRAMs angegeben.

(siehe auch Datenblatt)

Bemerkung zum Korrektheitsbeweis (1): Schaltbild eines SRAMs

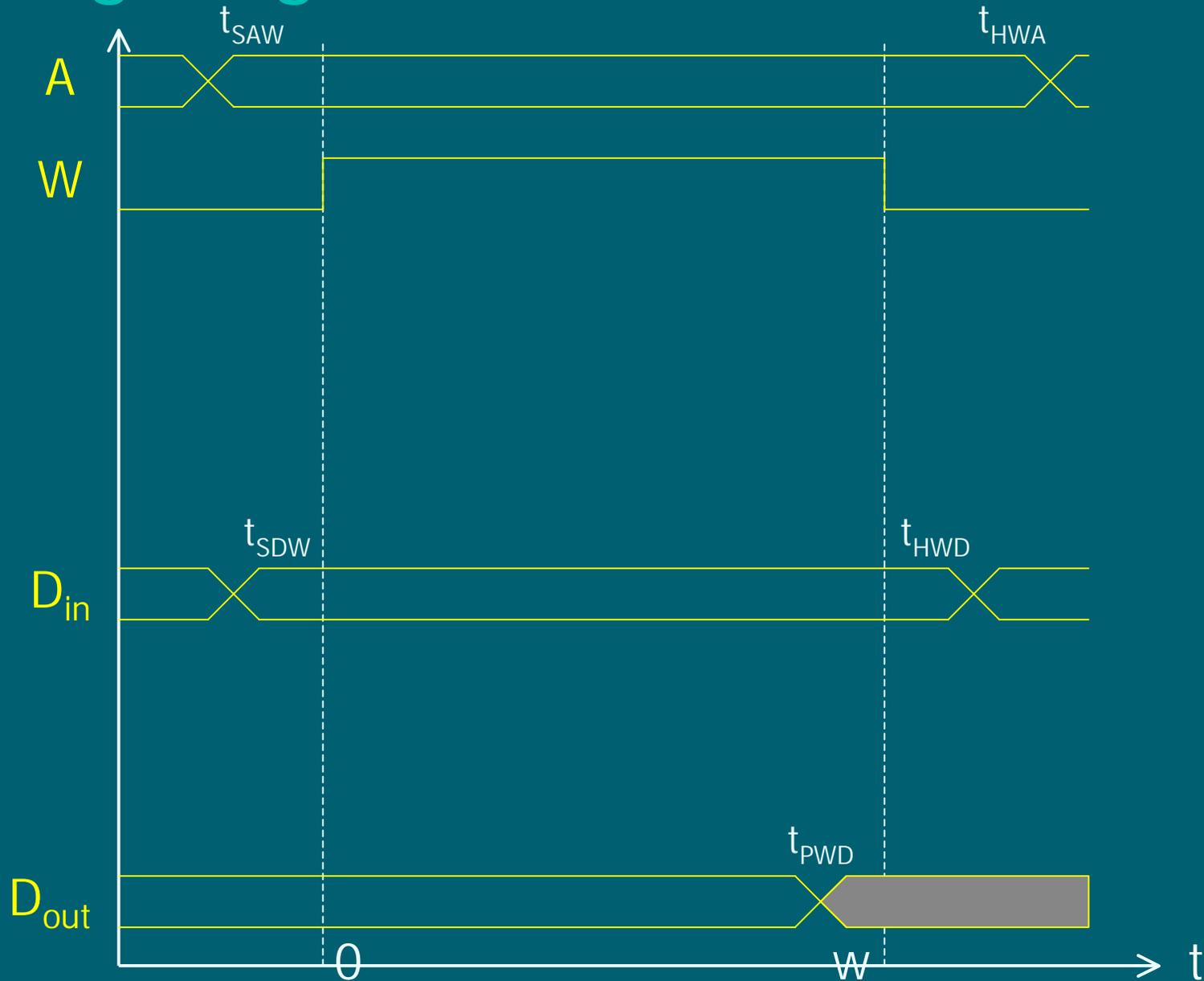


Bemerkung zum Korrektheitsbeweis (2):

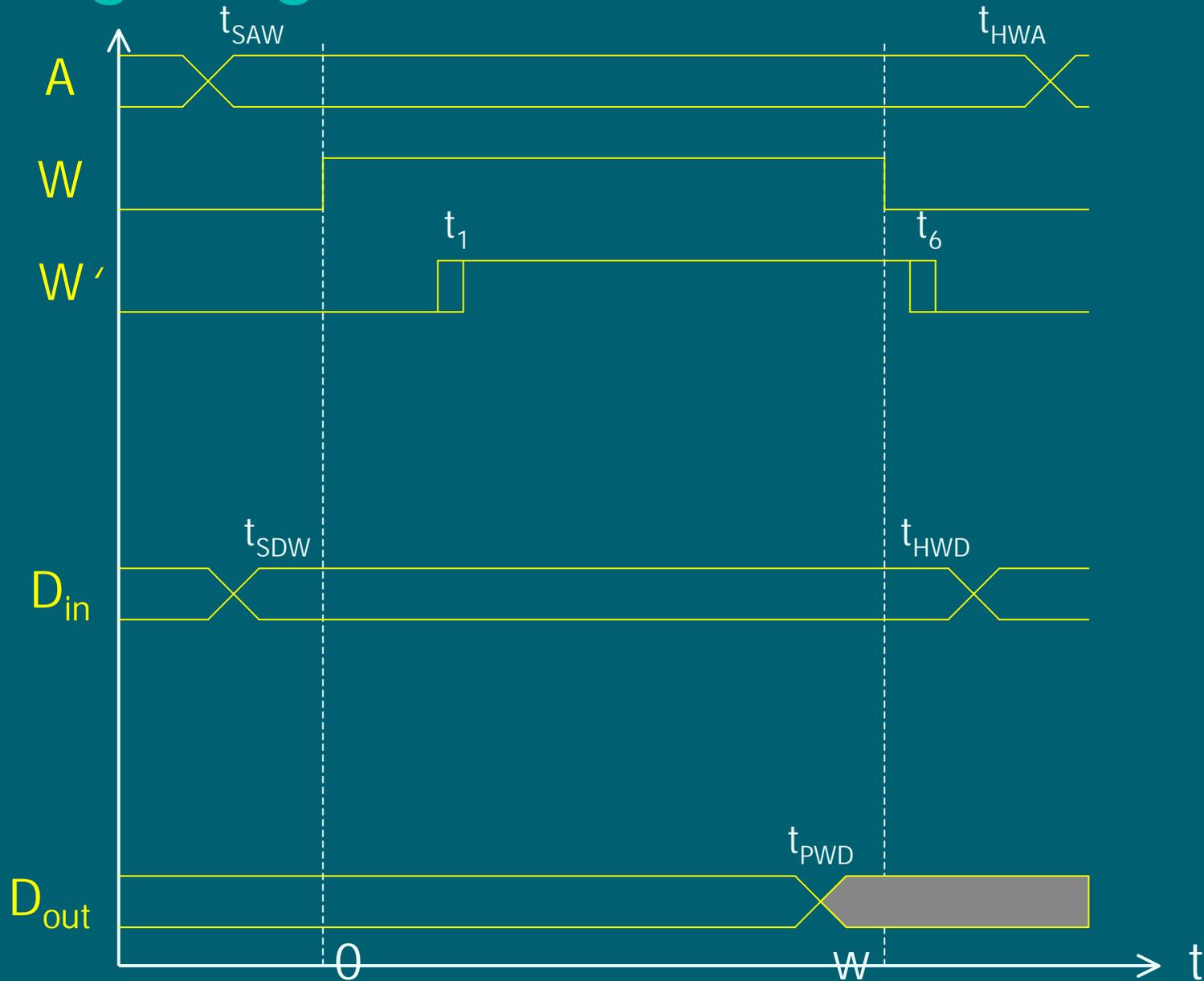
- I. Schreibpulsweite für D-Latch muss ausreichend sein.
- II. Auf Signalen W_i (*individuelle Schreibsignale*) dürfen keine Spikes entstehen.
- III. Setup-Zeiten der D-Latches sind zu erfüllen.
- IV. Hold-Zeiten der D-Latches sind zu erfüllen.



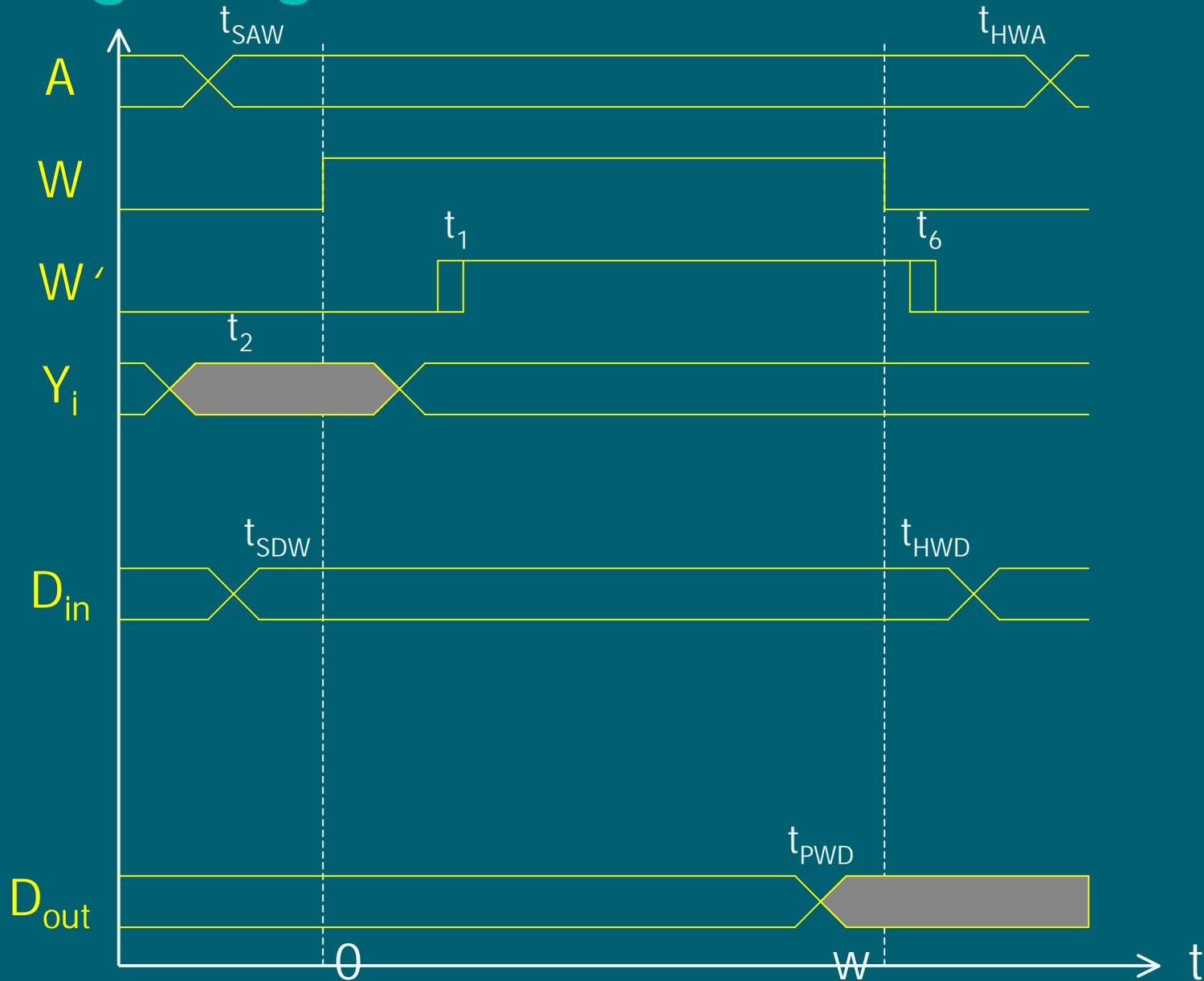
Bemerkung zum Korrektheitsbeweis (3): Timing-Diagramm



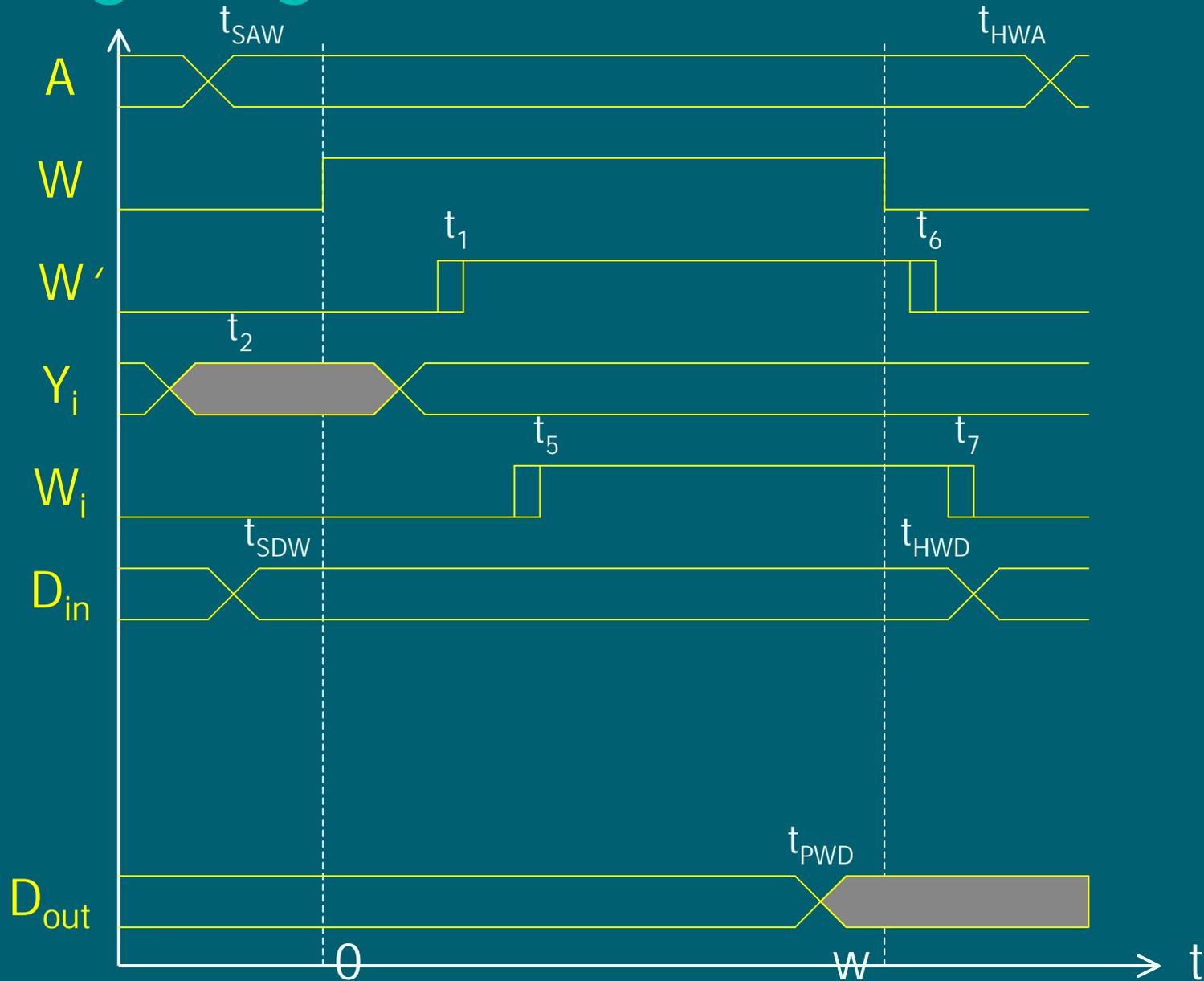
Bemerkung zum Korrektheitsbeweis (3): Timing-Diagramm



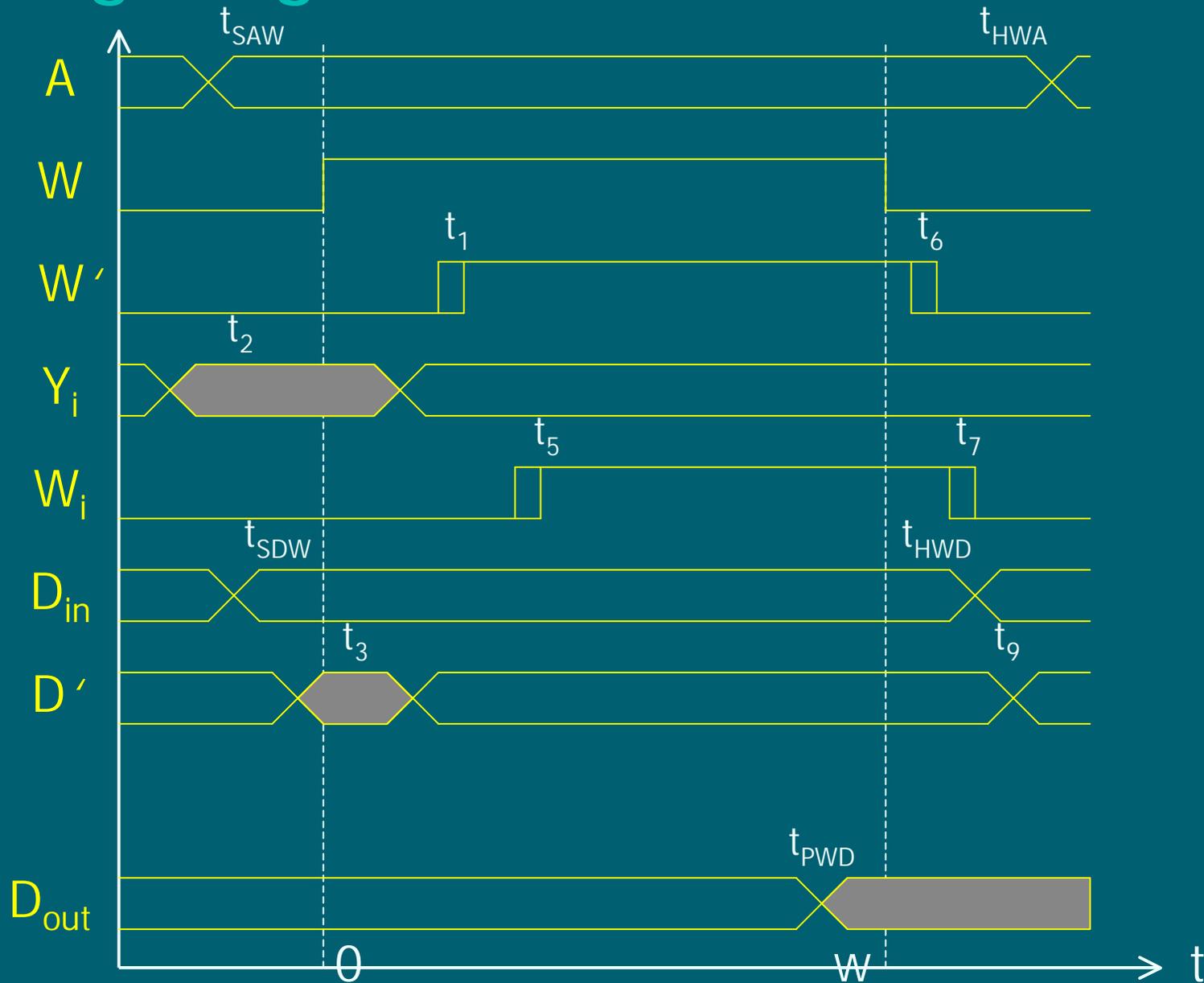
Bemerkung zum Korrektheitsbeweis (3): Timing-Diagramm



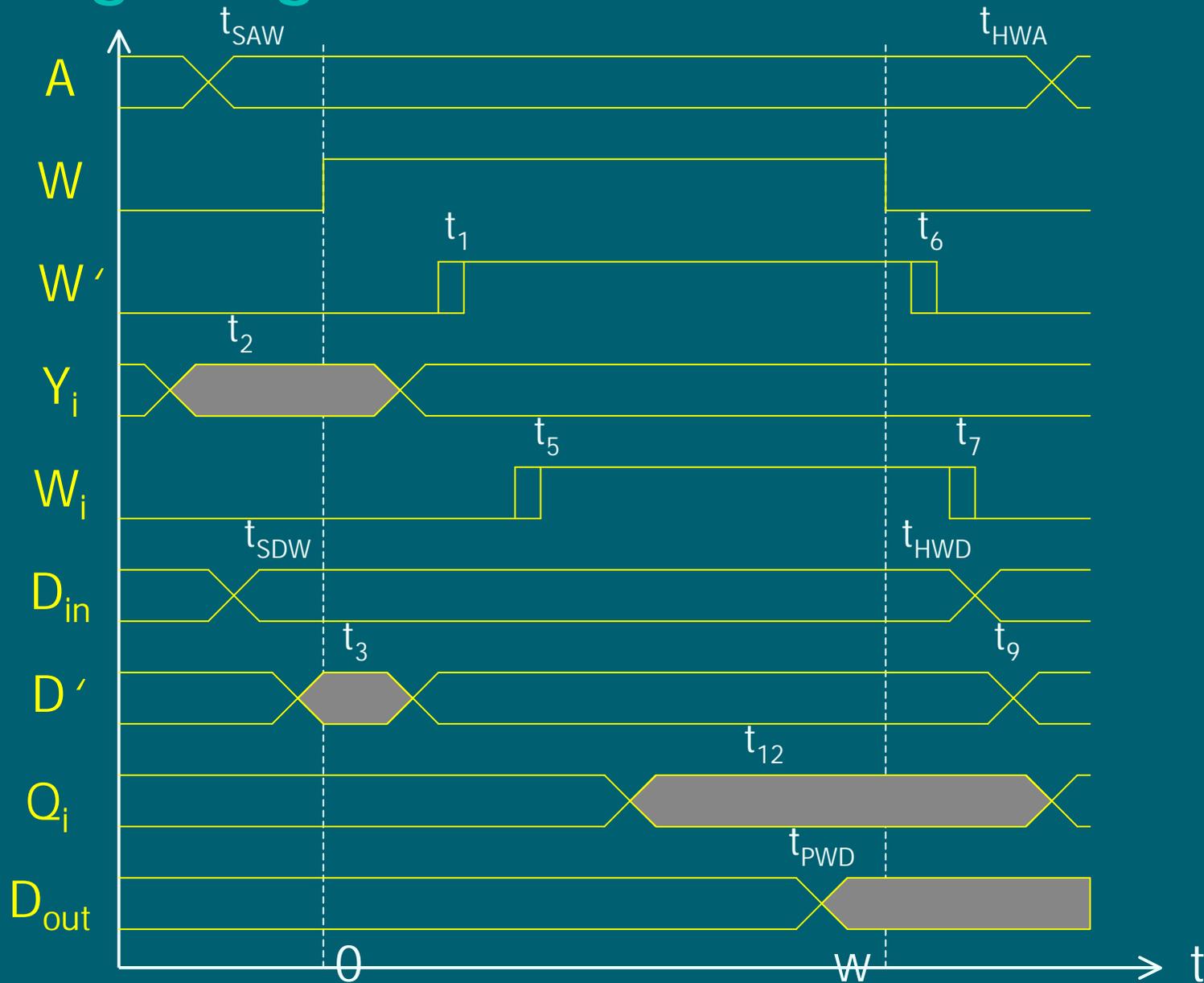
Bemerkung zum Korrektheitsbeweis (3): Timing-Diagramm



Bemerkung zum Korrektheitsbeweis (3): Timing-Diagramm



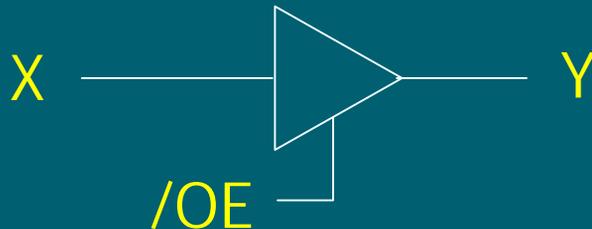
Bemerkung zum Korrektheitsbeweis (3): Timing-Diagramm



Zum Schluss von 12.3: Tristate-Treiber und Busse

Tristate-Treiber sind Treiber mit Eingangssignal X und zusätzlichem Signal $/OE$, dem Output Enable – Signal

Schaltsymbol:



Tristate-Treiber (ff)

Am Ausgang Y erscheint

$$Y = \begin{cases} X, & \text{falls } /OE = 0 \\ Z, & \text{falls } /OE = 1 \end{cases}$$

Z bezeichnet einen Zustand hoher Impedanz
(high Z, Z)

Tristate-Treiber (ff)

Ein Tristate-Treiber, dessen Ausgang mit Leitung I verbunden ist und sich im Zustand Z befindet, verhält sich so, als sei die Verbindung zu I unterbrochen!

Man sagt:

„ *Der Treiber ist disabled.* “

(Gegensatz: *enabled*)

n-Bit-Treiber

n Treiber mit gemeinsamen /OE sind **n-Bit-Treiber**

Beispiel: 74F244 enthält 2 4-Bit-Treiber

Im Gegensatz zu Ausgängen üblicher Gatter kann man Ausgänge von Tristate-Treibern **zusammenschalten**.

Man muss dafür sorgen, dass zu jeder Zeit **höchstens ein Treiber enabled** ist.

Busse

Ein **Bus** ist eine Leitung,
die mehrere Treiberausgänge verbindet.



n-Bit breiter Bus =

n-Bit Treiber mit n Leitungen verbunden

Zeitverhalten von Tristate-Treibern

Enable-Zeit $t_{pZX} =$

Zeit vom Aktivieren von /OE bis zum Durchschalten

Disable-Zeit $t_{pXZ} =$

Zeit vom Deaktivieren von /OE bis zum Isolieren
des Ausgangs

Zeitangaben zu Treibern

	Treiber 74F244	min	max
t_{PZL}	Enable-Zeiten	2.0	8.0
t_{PZH}	Enable-Zeiten	2.0	6.7
t_{PLZ}	Disable-Zeiten	2.0	7.0
t_{PHZ}	Disable-Zeiten	2.0	7.0
t_{PLH}	Umschaltverzögerung bei /OE = 0	2.5	6.2
t_{PHL}	Umschaltverzögerung bei /OE = 0	2.5	6.5

Vergleich: Bus vs. Mux

k Tristate-Treiber, die durch einen Bus verbunden sind, wirken ähnlich wie ein k -fach-Multiplexer.

Vorteile Bus gegenüber Multiplexer:

- leicht erweiterbar
(siehe Bus im PC mit Steckkarten)
- Datentransport in verschiedene Richtungen zu verschiedenen Zeiten

Problem

Vermeidung von **Bus Contention**

d.h. nie 2 Treiber auf einem Bus gleichzeitig enablen!!



Beispiel zur Verwendung

SRAM mit gemeinsamem Dateneingang und -ausgang:

